AT8F2481

22 I/O 22+2-ch ADC 8-bit Flash-Based MCU

Version 1.0

Nov. 28, 2025

Revision History

Version	*Date*	*Description*	*Modified Page*
1.0	2025/11/28	Formal release.	-

Table of Contents

1. 產品特性

1.1 功能特性

- 指令系統
  - 系統時鐘最高支持 20MHz/2T
- 內存
  - 4Kx16 bits Flash。
  - 336 bytes SRAM。
  - 256 bytes EEPROM。
- 5 種震盪方式
  - I_HRC-內部高速震盪: 最高 24MHz
  - E_HXT-外部高速震盪: 最高 (24MHZ)
  - E_XT-外部震盪:455K~6MHz
  - I_LRC-內部低速震盪:32.768KHz
  - E_LXT-外部低速震盪: 32.768KHz
- 內置WDT 定時器
- 內建電阻頻率轉換器(RFC)功能.
- 中斷源
  - PA / PB / PC口電平變化中斷
  - 外部中斷INT0 / INT1 / INT2
  - 其他中斷
- 定時器
  - 8 位定時器Timer0 可擇外部 32.768KHz或內部 32.768KHz。
  - 10 位定時器 Timer1 / Timer4 / Timer5
- 低壓復位功能(LVR)
  - 1.8V / 2.0V / 2.2V / 2.4V / 2.7V / 3.0 / 3.3V
- GPIO
  - 最多 22 個GPIO
  - 所有數字功能可分配到任意GPIO
  - 均支持上/下拉電阻功能
- 低電壓檢測(LVD)
  - 內建 16 階準確的低電壓偵測電路功能
- 通信模塊
  - 1xSPI
  - 1xI^2C
  - 1xUART
- 工作電壓範圍:
  - 2.7V~5.5V @ 20MHz/2T
  - 1.8V~5.5V @ 4MHz/4T
- 工作溫度
  - -40℃~85℃
- 蜂鳴器驅動
  - 50%佔空比，頻率可自由設置
- 增強型PWM
  - 五通道PWM(1 / 2 / 3 / 4 / 5)
  - 支持獨立/互補/同步/成組模式
  - 支持CCP 死區互補模式
  - 1 組全橋(FB)或 3 組半橋(HB)模式
- 高精度 12 位ADC
  - 所有GPIO(22I/Os)均支持AD通道
  - 參考電壓可選擇(2V / 3V / 4V / VDD)
  - 可支持外部電壓檢測(PA0 或PB1)
  - 支持硬件觸發自動轉換功能
- 一路模擬比較器
  - 正端選擇內部Vref 或外部電壓(PA0 / PA2 / PA4)
  - 負端可選擇內部 0.6V基準電壓或Vref或 (PA1 / PA3 / PB3)
- 兩路運算放大器(OPA0 / OPA1)
  - 每個運放三端均和GPIO端複用
  - 支持運算放大器/比較器兩種模式
  - 運放輸出可接內部ADC通道
  - 支持失調電壓軟件修調
- 內建變頻振盪器(V_HRC)
  - 提供四種選擇: 32MHz / 20.8MHz / 16MHz / 13.6MHz。
- 內建二線在電路燒寫(ICP)
- 內建二線控制的除錯仿真電路 (On-Chip Debug)(OCD)。
- 支持二線帶電程序升級(In-System Program)(ISP)。

1. General Description

1.1 Features

- Instruction System clock per instruction, System clock supports up to 20 MHz with 2T architecture for efficient processing.
- Memory
  - 4K x 16 bits Flash.
  - 336 bytes SRAM.
  - 256 bytes EEPROM
- Oscillator Options (5 types)
  - **I_HRC**: Internal High-Speed Oscillator (24MHz).
  - **E_HXT**: External High-Speed Crystal (up to 24MHz).
  - **E_XT**: External Oscillator (455 KHz to 6 MHz).
  - **I_LRC**: Internal Low-Speed Oscillator (32.768 KHz).
  - **E_LXT**: External Low-Speed Crystal (32.768 KHz).
- Built-in Watch-Dog Timer (WDT)
- Built-in Resistance to Frequency Converter (RFC) function.
- Timer Functions
  - 8-bit Timer0: Selectable internal or external 32.768 kHz as source
  - 10-bit Timers: Timer1 / Timer4 / Timer5
- Interrupt Sources.
  - GPIO level or edge-triggered interrupts (Ports PA/PB/PC).
  - External interrupts: INT0/INT1/INT2
  - Other internal interrupt sources
- Low Voltage Reset (LVR)
  - Selectable thresholds

  1.8 V / 2.0 V / 2.2V / 2.4 V / 2.7 V / 3.0 V / 3.3 V
- Low Voltage Detection (LVD)
  - Built-in 16-level accurate low voltage detection circuit function
- Buzzer Driver

  Fixed 50% duty cycle.

  Output frequency freely configurable.
- Operating voltage range
  - 2.7V ~ 5.5V @ 20MHz/2T
  - 1.8V ~ 5.5V @ 4MHz/4T
- Operating temperature range
  - -40℃ to 85℃.
- GPIO
  - Up to 22 General Purpose I/O pins.
  - All digital functions can be mapped to any GPIO.
  - Integrated pull-up / pull-down resistor support.
- Communication Interfaces.
  - 1 × SPI.
  - 1 × I²C.
  - 1 × UART.
- Enhanced PWM
  - 5 Channels PWM: PWM1 / PWM2 / PWM3 / PWM4 / PWM5.
  - Supports independent, complementary, synchronous, and group modes.
  - CCP mode supports complementary with dead-time.
  - Supports 1 full-bridge (FB) or 3 half-bridge (HB) output configurations
- High-Precision 12-bit ADC
  - All GPIOs (up to 22 I/Os) can be configured as ADC channels.
  - Selectable reference voltage: 2V / 3V / 4V / VDD.
  - Supports external voltage sensing on PA0 or PB1.
  - Supports auto conversion by hardware trigger
- 1-channel Comparator

The positive input can select either internal Vref or external voltage sources (PA0 / PA2 / PA4).

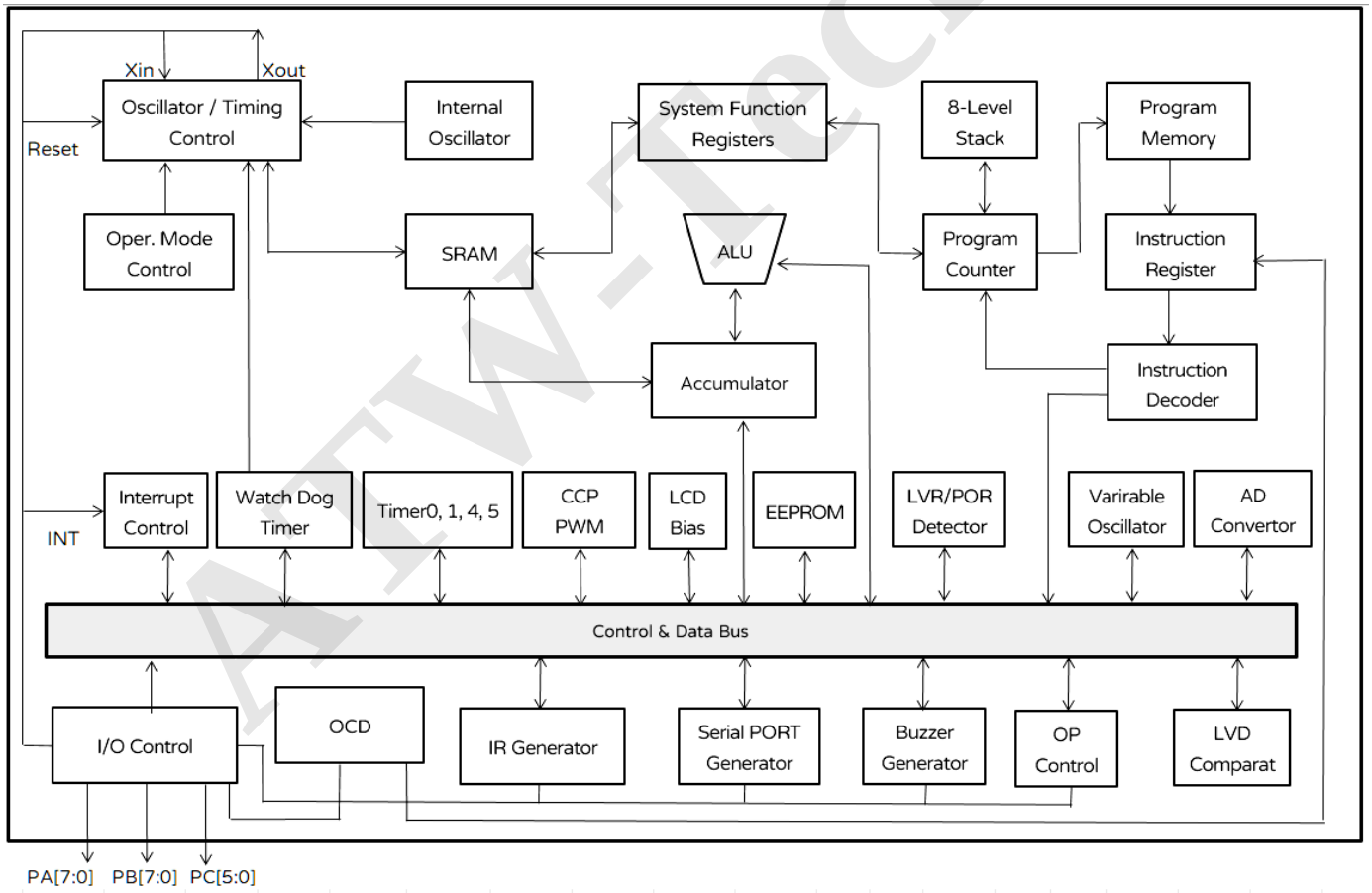
The negative input can select either the internal reference voltage 0.6V or Vref or (PA1 / PB3 / PA3).

- Operational Amplifiers (OPA0 / OPA1)
  Each op-amp's three terminals are multiplexed with GPIO pins.
  Supports two operation modes: Operational Amplifier / Comparator
  Op-amp output can be connected to internal ADC channels.
  Input offset voltage can be adjusted via software.

- Built-in Variable High-Speed Oscillator (**V_HRC**)
  Provides four selectable frequencies: 32MHz / 20.8MHz / 16MHz / 13.6MHz.
- Support In-Circuit Programming(ICP)
- Built-in Two-wire On-Chip Debug Circuit(OCD)
  Supports program download and real-time debugging during development.
- Support In-System Program(ISP)

.

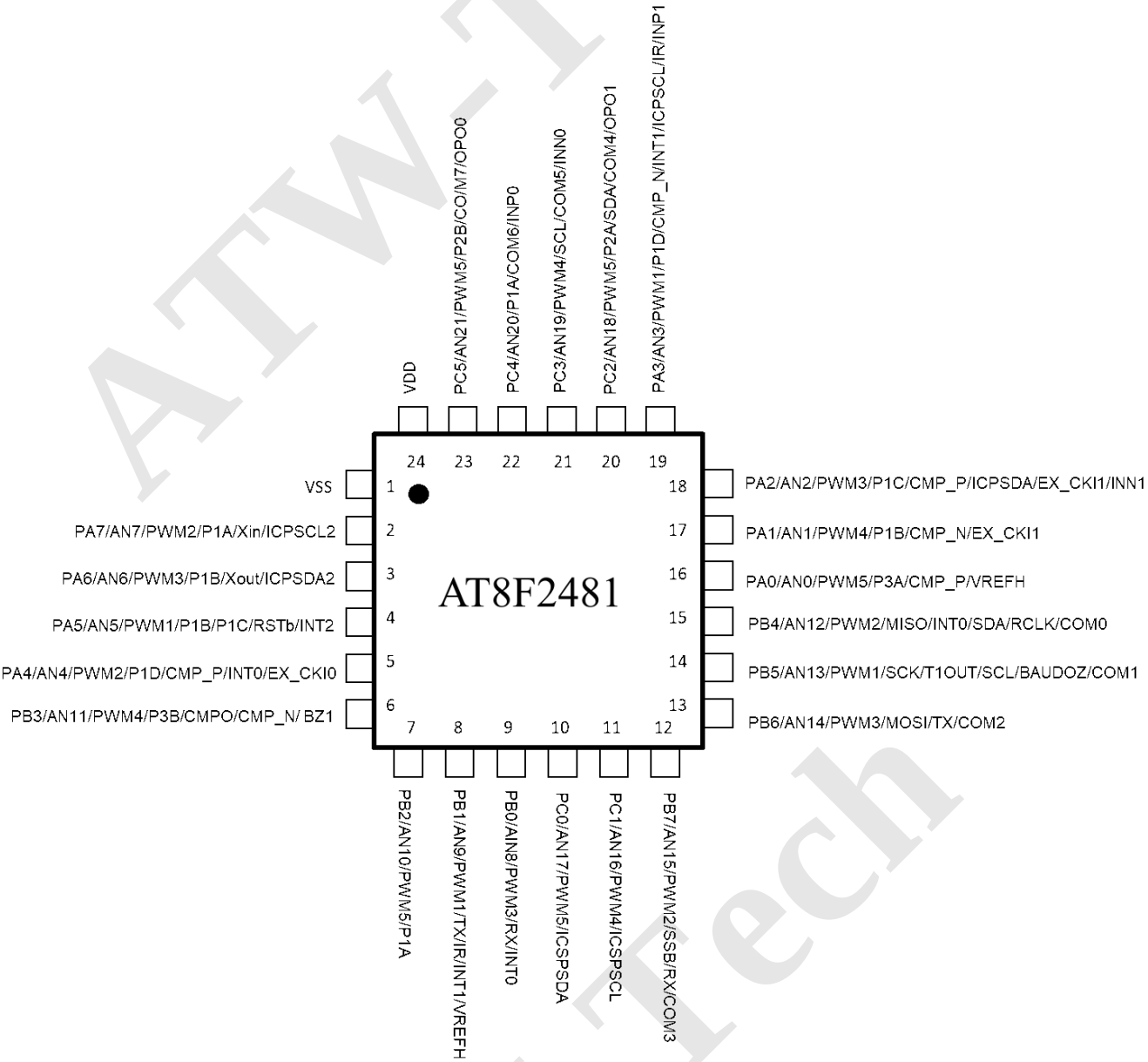
1.2 Block Diagram



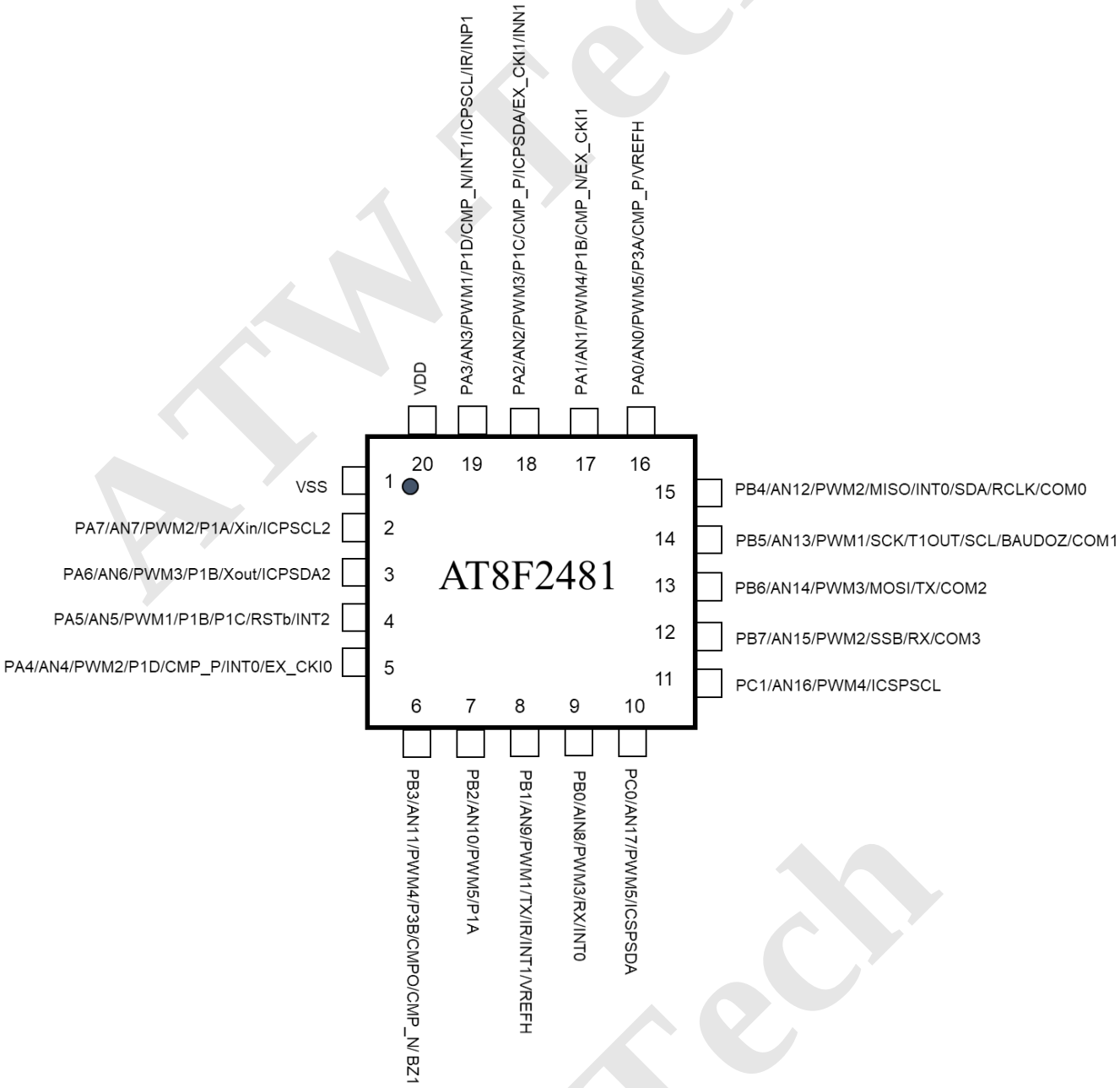
1.3 Pin Assignment

AT8F2481 provides the following package types: QFN24, QFN20, SSOP24, TSSOP20, SOP16, and SOP8.

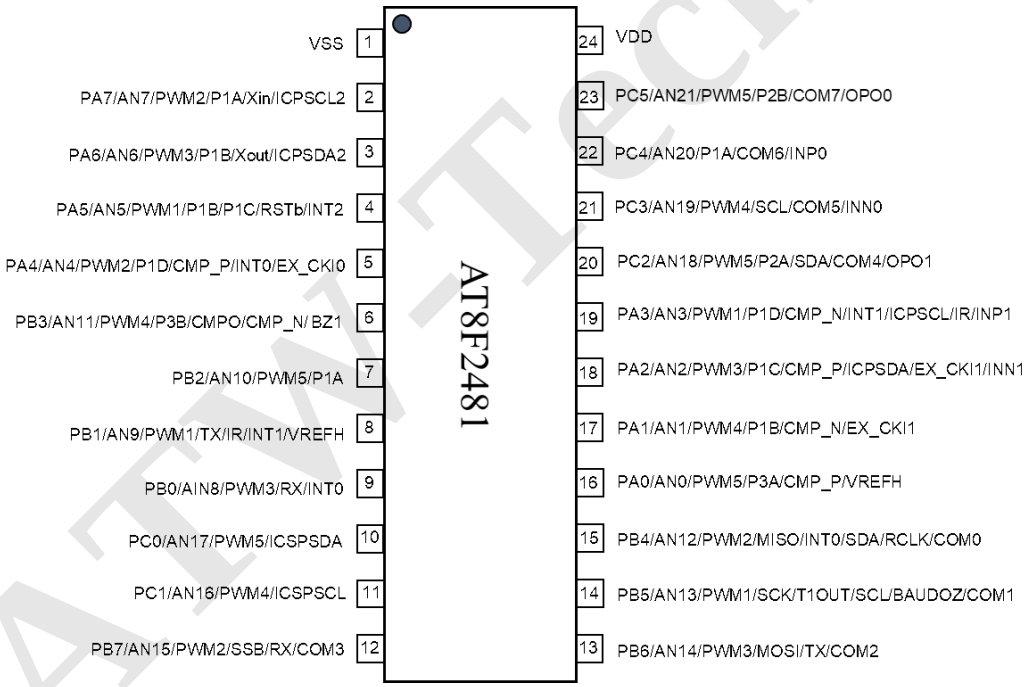
QFN24 package



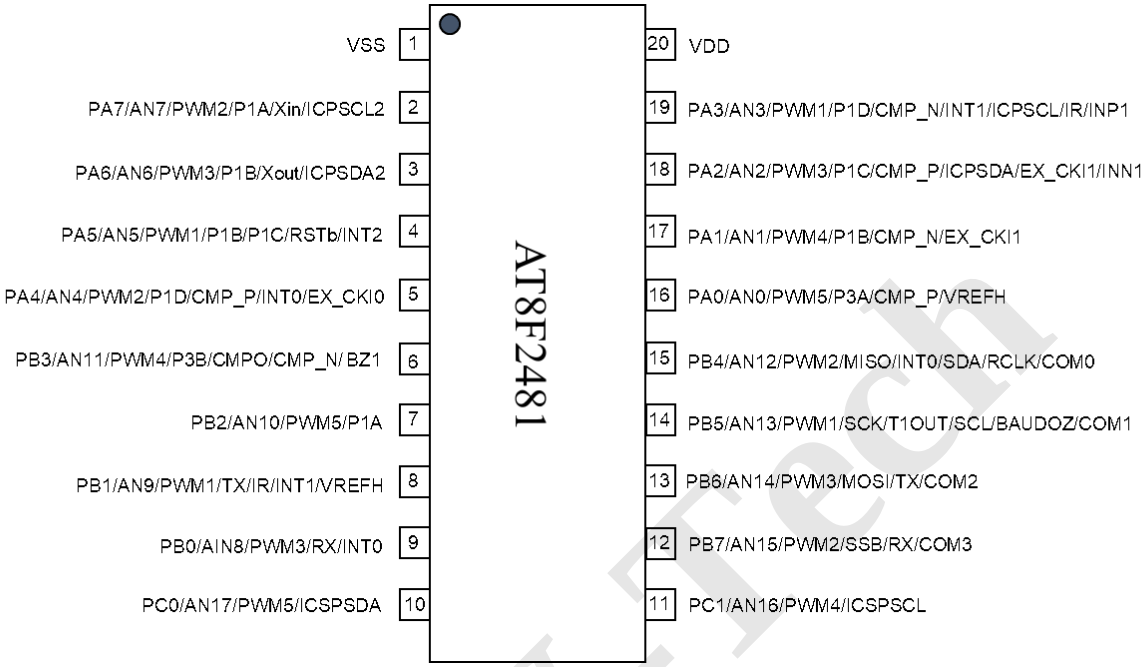
QFN20 package



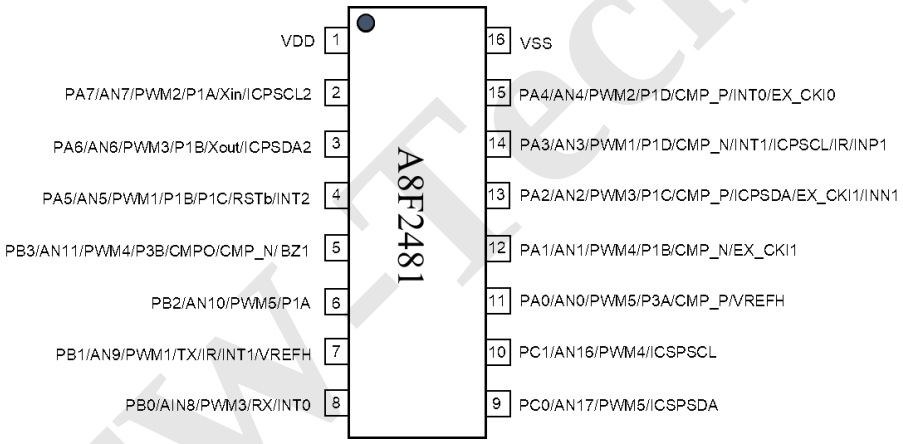
SSOP24 Package



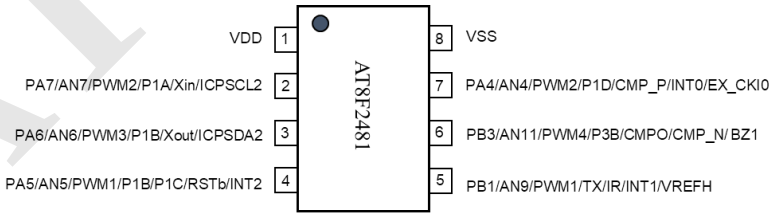
TSSOP20 Package



SOP16 Package



SOP8 Package



1.4 Pin Description

Pin Name	I/O	Description
PA0 / AN0 / PWM5 / P3A / CMP_P / VREFH	I/O	PA0 is bidirectional I/O pin. AN0 is ADC analog input pin. PWM5 is PWM output pin P3A is CCP output pin. CMP_P is comparator positive input pin. VREFH is ADC high reference input pin.
PA1 / AN1 / PWM4 / P1B / CMP_N / EX_CKI1	I/O	PA1 is bidirectional I/O pin. AN1 is ADC analog input pin. PWM4 is PWM output pin. P1B is CCP output pin. CMP_N is comparator negative input pin. EX_CKI1 is Timer4/5 clock source.
PA2 / AN2 / PWM3 / P1C / CMP_P / ICPSDA / EX_CKI1 / INN1	I/O	PA2 is bidirectional I/O pin. AN2 is ADC analog input pin. PWM3 is PWM output pin. P1C is CCP output pin. CMP_P is comparator positive input pin. ICPSDA is ICP/OCD SDA pin. EX_CKI1 is Timer4/5 clock source. OPA1 negative node input

Pin Name	I/O	Description
PA3 /	I/O	PA3 is bidirectional I/O pin.
AN3 /		AN3 is ADC analog input pin.
PWM1 /		PWM1 is PWM output pin.
P1D /		P1D is CCP output pin.
CMP_N /		CMP_N is comparator negative input pin.
INT1 /		INT1 is external interrupt pin.
ICPSCL /		ICPSCL is ICP/OCD SCL pin.
IR /		If IR mode is enabled, PA3 is IR carrier output with large sink current.
INP1		OPA1 positive node input
PA4 /	I/O	PA4 is bidirectional I/O pin.
AN4 /		AN4 is ADC analog input pin.
PWM2 /		PWM2 is PWM output pin.
P1D /		P1D is CCP output pin
CMP_P /		CMP_P is comparator positive input pin.
INT0 /		INT0 is external interrupt pin.
EX_CKI0		EX_CKI0 is Timer0/1 clock source.
PA5 /	I/O	PA5 is bidirectional I/O pin.
AN5 /		AN5 is ADC analog input pin.
PWM1 /		PWM1 is PWM output pin.
P1B /		P1B is CCP output pin.
P1C /		P1C is CCP output pin
RSTb /		PA5 can be the reset pin RSTb.
INT2		INT2 is external interrupt pin.
PA6 /	I/O	PA6 is bidirectional I/O pin.
AN6 /		AN6 is ADC analog input pin.
PWM3 /		PWM3 is PWM output pin.
P1B /		P1B is CCP output pin
Xout /		PA6 can be the output pin of crystal oscillator Xout.
ICPSDA2		ICPSDA2 is ICP SCL pin.
PA7 /	I/O	PA7 is bidirectional I/O pin.
AN7 /		AN7 is ADC analog input pin.
PWM2 /		PWM2 is PWM output pin.
P1A /		P1A is CCP output pin
Xin /		PA7 can be the input pin of crystal oscillator Xin.
ICPSCL2		ICPSCL2 is ICP SDA pin.
PB0 /	I/O	PB0 is bidirectional I/O pin.
AIN8 /		AN8 is ADC analog input pin.
PWM3 /		PWM3 is PWM output pin.
RX /		RX is UART RX pin.
INT0		INT0 is external interrupt pin .

Pin Name	I/O	Description
PB1 / AN9 / PWM1 / TX / IR / INT1 / VREFH	I/O	PB1 is bidirectional I/O pin. AN9 is ADC analog input pin. PWM1 is PWM output pin. TX is UART TX pin. If IR mode is enabled, PB1 is IR carrier output with large sink current。 INT1 is external interrupt pin. VREFH is ADC high reference input pin.
PB2 / AN10 / PWM5 / P1A	I/O	PB2 is bidirectional I/O pin. AN10 is ADC analog input pin. PWM5 is PWM output pin. P1A is CCP output pin.
PB3 / AN11 / PWM4 / P3B / CMPO / CMP_N / BZ1	I/O	PB3 is bidirectional I/O pin. AN11 is ADC analog input pin. PWM4 is PWM output pin. P3B is CCP output pin. CMPO is comparator output pin. CMP_N is comparator negative input pin. BZ1 is buzzer output pin.
PB4 / AN12 / PWM2 / MISO / INT0 / SDA / RCLK / COM0	I/O	PB4 is bidirectional I/O pin. AN12 is ADC analog input pin. PWM2 is PWM output pin. MISO is SPI output pin. INT0 is external interrupt pin. SDA is I^2C data pin. RCLK is UART Baud rate input pin. PB4 can output LCD 1/2 VDD COM0 voltage.
PB5 / AN13 / PWM1 / SCK / T1OUT / SCL / BAUDOZ / COM1	I/O	PB5 is bidirectional I/O pin. AN13 is ADC analog input pin. PWM1 is PWM output pin. SCK is SPI clock input pin. Timer1 match output pin, T1OUT toggles when Timer1 underflow occurs. SCL is I^2C clock input pin. BAUDOZ is baud rate clock output pin. PB5 can output LCD 1/2 VDD COM1 voltage.
PB6 / AN14 / PWM3 / MOSI / TX / COM2	I/O	PB6 is bidirectional I/O pin. AN14 is ADC analog input pin. PWM3 is PWM output pin. MOSI is SPI input pin. TX is UART TX pin. PB6 can output LCD 1/2 VDD COM2 voltage.

Pin Name	I/O	Description
PB7 / AN15 / PWM2 / SSB / RX / COM3	I/O	PB7 is bidirectional I/O pin. AN15 is ADC analog input pin. PWM2 is PWM output pin. SSB is SPI enable pin. RX is UART RX pin. PB7 can output LCD 1/2 VDD COM3 voltage.
PC0 / AN17 / PWM5 / ICSPSDA	I/O	PC0 is bidirectional I/O pin. AN17 is ADC analog input pin. PWM5 is PWM output pin. ICSPSDA is ICP/OCD/ISP data pin.
PC1 / AN16 / PWM4 / ICSPSCL	I/O	PC1 is bidirectional I/O pin. AN16 is ADC analog input pin. PWM4 is PWM output pin. ICSPSCL is ICP/OCD/ISP clock pin.
PC2 / AN18 / PWM5 / P2A / SDA / COM4 / OPO1	I/O	PC2 is bidirectional I/O pin. AN18 is ADC analog input pin. PWM5 is PWM output pin. P2A is CCP output pin. SDA is I^2C SDA pin. PC2 can output LCD 1/2 VDD COM4 voltage. OPA1 output node
PC3 / AN19 / PWM4 / SCL / COM5 / INN0	I/O	PC3 is bidirectional I/O pin. AN19 is ADC analog input pin. PWM4 is PWM output pin. SCL is I^2C SCL pin. PC3 can output LCD 1/2 VDD COM5 voltage. OPA0 negative node input
PC4 / AN20 / P1A / COM6 / INP0	I/O	PC4 is bidirectional I/O pin. AN20 is ADC analog input pin. P1A is CCP output pin. PC4 can output LCD 1/2 VDD COM6 voltage. OPA0 positive node input
PC5 / AN21 / PWM5 / P2B / COM7 / OPO0	I/O	PC5 is bidirectional I/O pin. AN21 is ADC analog input pin. PWM5 is PWM output pin. P2B is CCP output pin. PC5 can output LCD 1/2 VDD COM voltage. OPA0 output node
VDD	P	Positive Power Supply.
VSS	P	Ground.

P=Power; I/O= Bidirectional;

1.5 In-Circuit Program (ICP)

1.5.1 Overview

The FLASH ROM based memory can be programmed by “In-Circuit Program” (ICP).

The ICP system uses a two-wire serial interface, ICPSCL and ICPSDA, to establish communication between the target device and the controlling programming host.

ICPSDA is the data input and output pin and ICPSCL is the clock input pin, which synchronizes the data shifted in or out from AT8F2481 under programming.

ATW provides ICP tool (Q-Writer) for AT8F2481, which enables user to easily perform ICP through ATW programmer MCU_Writer.

1.5.2 Limitation of ICP

The ICPSCL and ICPSDA pins are physically located on three group pins, which are PC1/PC0 or PA3/PA2 or PA7/PA6.

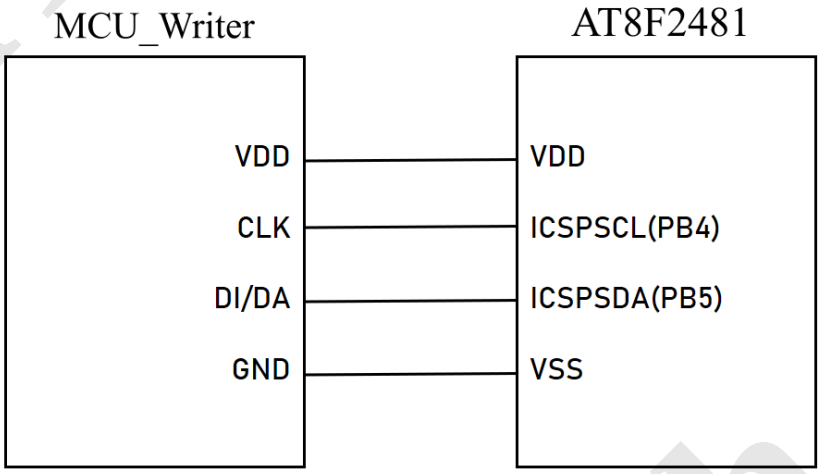


Figure 1 Typical Connection for ICP Programming

1.6 On-Chip Debug (OCD)

1.6.1 Overview

AT8F2481 is embedded in an On-Chip Debugger(OCD) providing developers with a low cost method for debugging user code. The OCD gives debug capability of complete program flow control with 3 hardware address breakpoints, 1 conditional register break, single step, free running, and non-intrusive commands for memory access.

The OCD system uses a two-wire serial interface, ICPSCL and ICPSDA, to establish communication between the target device and the controlling debugger host. ICPSDA is an input/output pin for debug data transfer and ICPSCL is an input pin for synchronization with ICPSDA. The AT8F2481 also use ICPSCL and ICPSDA as control pins to write and read it.

AT8F2481 shares the ROM address 0xF00~0xFFF as OCD space. If user use more than 0xEFF program space, OCD function would be disabled.

1.6.2 Limitation of OCD

AT8F2481 is a fully-featured microcontroller that multiplexes several functions on its limited I/O pins. Some device functionality must be sacrificed to provide resources for OCD system. The OCD has the following limitations:

1. The OCD CLK/data pins are physically located on the same pin PC1/PC0 or PA3/PA2, but not including PA6/PA7.

Therefore, neither its I/O function nor shared multi-functions can be emulated.

2. System clock cannot be turned off because OCD uses this clock to monitor its internal status: When the system is in halt mode, it is invalid to perform ram/registers accesses because parts of the device may not be clocked. A read access could return garbage or a write access might not succeed. But the following accesses are not affected by the system halt: Read current program address, current PCL, current break condition and current halt status.

1.7 In-System Program (ISP)

1.7.1 Overview

The FLASH ROM based memory can be programmed by “In-System Program” (ISP). If the product is the end product needs firmware updating in the hand of an end customer, the hardware programming mode will make repeated programming difficult and inconvenient, ISP method make it easy and possible without removing the AT8F2481 from the system.

There are two signal pins ICSPSDA and ICSPSCL, involved in ISP function. ICSPSDA is the data input and output pin. ICSPSCL is the clock input pin, which synchronizes the data shifted in or out from AT8F2481 under programming. User should leave these two pins plus VDD and VSS pins on the circuit board to make ISP possible.

ATW provides ISP tool (NYISP) for AT8F2481, which enables user to easily perform ISP through ATW programmer.

1.7.2 Limitation of ISP

To avoid current leakage, the two signal pins ICSPSDA and ICSPSCL, must be set as input pull high I/O. ISP mode does not allow customers to manufacture circuit boards with un-programmed devices

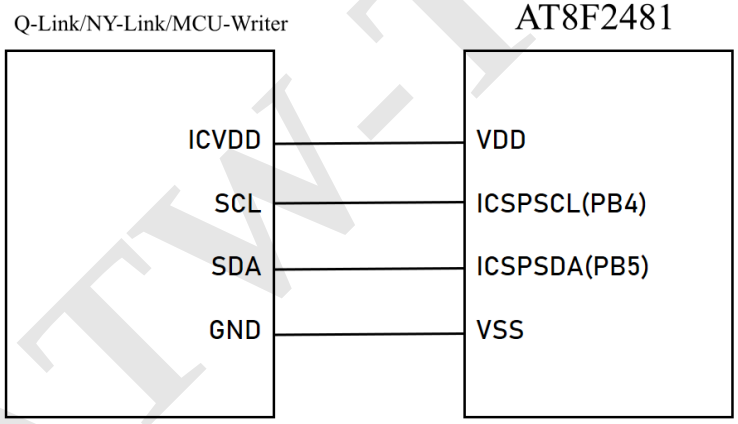


Figure 2 Typical Connection for ISP Programming

2. Memory Organization

AT8F2481 memory is divided into two categories: one is program memory and the other is data memory.

2.1 Program Memory

The program memory space of AT8F2481 is 4K words. Therefore, the Program Counter (PC) is 12-bit wide in order to address any location of program memory.

Some locations of program memory are reserved as interrupt entrance. Power-On Reset vector is located at 0x000. Internal and external hardware interrupt vector is located at 0x004.

AT8F2481 provides instruction LGOTO, LCALL to address 4K location of program space. (Note: In enhanced instruction set version, LGOTO and LCALL instruction have the ability to address any location of program memory in 4K words in an instruction words)

When a call instruction is executed or an interrupt causes a branch, next ROM address is written to top of the stack, when RET or RETIE instructions are executed, the top of stack data is loaded to PC.

AT8F2481 share the ROM address 0xF00~0xFFF as OCD space. If user use more than 0xEFF program space, OCD function would be disabled.

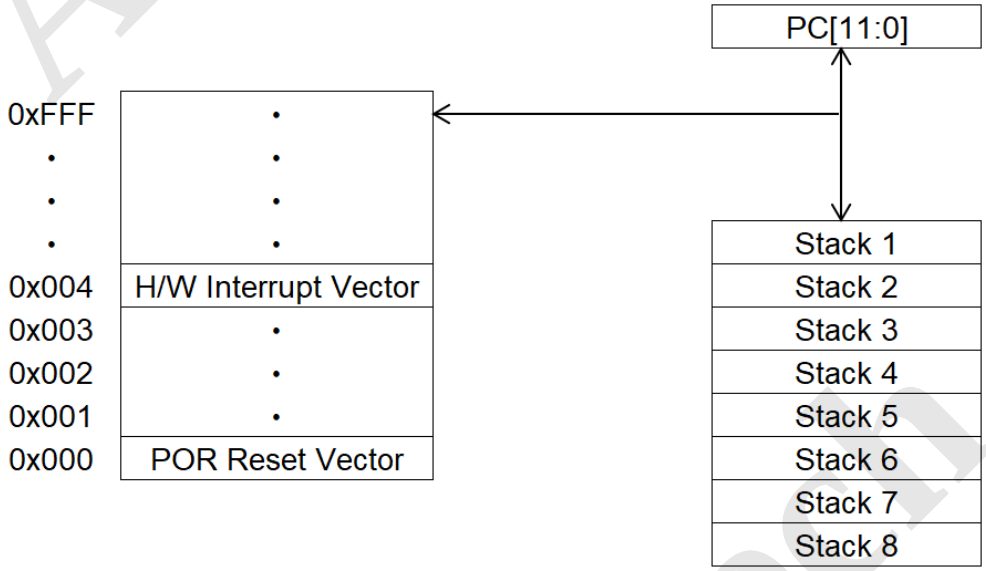


Figure 3 Program Memory Address Mapping

2.1.1 Program Memory Register

2.1.1.1 PCL (Low Byte of PC[11:0])

The register PCL is the least significant byte (LSB) of 12-bit PC. PCL will be increased by one after instruction is executed except some instructions which will change PC directly. The high byte of PC, i.e. PC[11:8], is not directly accessible. Update of PC[11:8] must be done through register PCHBUF.

When Write to PCL, PC[7:0] is the written data. PC[11:8] is from PCHBUF[3:0].

For LGOTO instruction, PC[11:0] is from instruction word.

For LCALL instruction, PC[11:0] is from instruction word.

Moreover, the next PC address, i.e. PC+1, will push onto top of Stack.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	0x2	PCL[7:0]							
R/W Property		R/W							
Initial Value		00000000							

2.1.1.2 PCHBUF (High Byte of PC)

When Write to PCL, PC[11:8] is from PCHBUF[3:0]. For LGOTO / LCALL instruction, PC[11] is from PCHBUF[3].

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	0xA	-	-		-	PCHBUF[3:0]			
R/W Property		-	-	-	-	W			
Initial Value		0	0	0	0	0000			

Bit 7:4 **Unimplemented**

Bit 3:0 **PCHBUF[3:0]:** Buffer of the 11^{th} ~ 8^{th} bit of PC

2.2 Data Memory

The Data memory is partitioned into 4 banks which contain the Special Function Register (SFR) and General Purpose Register (GPR). The SFR are located in the first 32 locations of each bank. The GPR, implemented as SRAM type, are located in the last 96 locations of each bank. The last 16 locations of Bank 1/2/3 are not implemented made the maximum number of GPR in AT8F2481 is 336.

Data memory has two addressing mode: direct addressing mode and indirect addressing mode.

The indirect addressing mode of data memory access is shown in the following figure. This indirect addressing mode is implied by accessing register INDF. That is, when accessing INDF SFR, the bank selection is determined by IRP(STATUS[7]) and FSR[7], the location selection is from FSR[6:0].

The direct addressing mode of data memory access is described below. The bank selection is determined by instruction op-code[15:14] immediately, and the location selection is from instruction op-code[6:0] immediately.

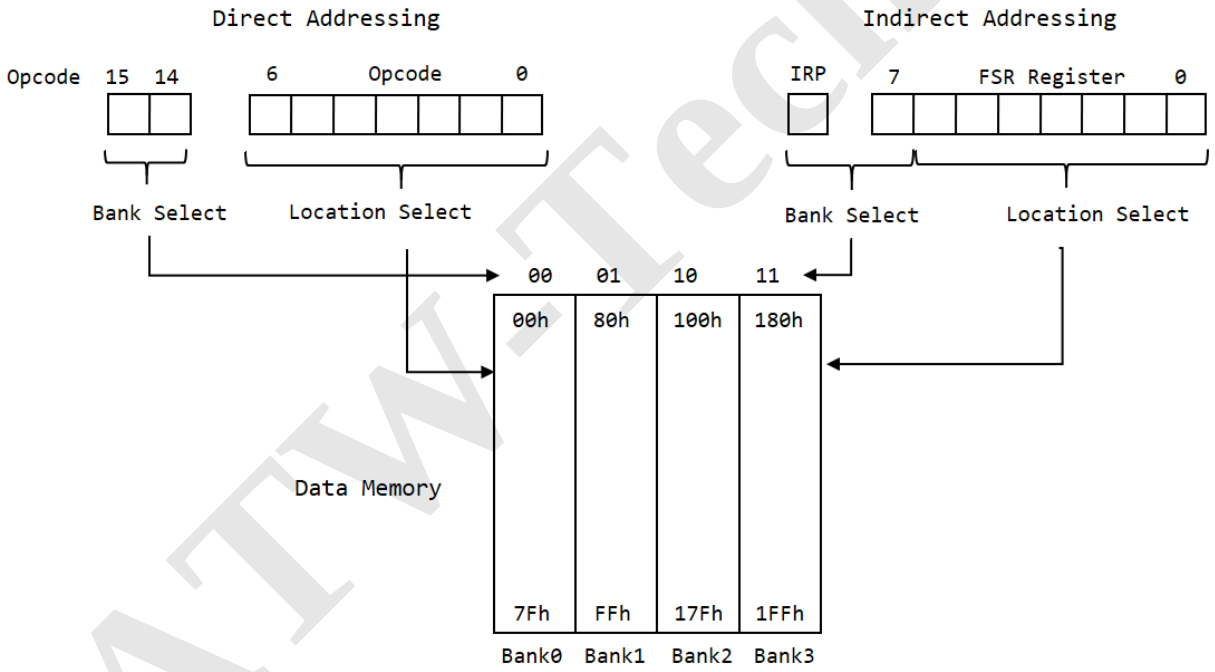


Figure 4 Indirect Addressing / Direct Addressing of Data Memory Access

```
; Write Data Memory by Direct Addressing Mode
MOVIA     0xA5
MOVAR     0x1A0          ; Write Data 0xA5 to Bank3 0x20
; Write Data Memory by Indirect Addressing Mode
BSR       STATUS,7       ; Select Bank3 by Set STATUS[7] and Set FSR[7]
MOVIA     0x80 | 0x20
MOVAR     FSR            ; FSR[6:0] Access Address 0x1A0(Bank3 0x20)
MOVIA     0xA5
MOVAR     INDF           ; Write Data 0xA5 toBank3 0x20
```

Data memory can be accessed by general instructions like arithmetic instructions and data movement instructions. The SFR occupies address from 0x0 to 0x1F of Bank 0~3. The GPR physically occupy address from 0x20 to 0x7F of Bank 0 and 0x20 to 0x6F of Bank 1~3. Access address 0x70~0x7F of Bank 1~3 will actually access to address of 0x70~0x7F of bank 0 instead.

The AT8F2481 register name and address mapping of SFR are described in the following table 2-1

2.2.1 Data Memory List

Address	Name	Address	Name	Address	Name	Address	Name
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	T0MD	101h	TMR0	181h	T0MD
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	IOSTA	105h	PORTA	185h	IOSTA
06h	PORTB	86h	IOSTB	106h	PORTB	186h	IOSTB
07h	PORTC	87h	IOSTC	107h	PORTC	187h	IOSTC
08h	Reserved	88h	Reserved	108h	Reserved	188h	OPACON
09h	OSCCR	89h	LVDCON	109h	ADMD	189h	RFC
0Ah	PCHBUF	8Ah	PCHBUF	10Ah	PCHBUF	18Ah	PCHBUF
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	ADDL	18Ch	LCDCON0
0Dh	PIR2	8Dh	PIE2	10Dh	ADDH	18Dh	LCDCON1
0Eh		8Eh		10Eh	ADCON1	18Eh	OSCCALL
0Fh	TMRxL	8Fh	PCON	10Fh	ADJMD	18Fh	OSCCALH
10h	TMRxH	90h	IRCR	110h	PxCON	190h	
11h	TxCR1	91h	PWM1DUTYL	111h	ADCR	191h	
12h	TxCR2	92h	PWM1DUTYH	112h	SIMDR	192h	
13h	BZ1CR	93h	PWM2DUTYL	113h	SIMCR	193h	
14h	AWUCON	94h	PWM2DUTYH	114h	SPCR	194h	
15h	BCDWUCON	95h	PWM3DUTYL	115h	MADR	195h	
16h	PORTACON30	96h	PWM3DUTYH	116h	MFDR	196h	
17h	PORTACON74	97h	PWM4DUTYL	117h	MCR	197h	
18h	PORTBCON30	98h	PWM4DUTYH	118h	MSR	198h	
19h	PORTBCON74	99h	PWM5DUTYL	119h	DLL	199h	
1Ah	PORTCCON74	9Ah	PWM5DUTYH	11Ah	DLH	19Ah	EEA
1Bh	PORTCCON30	9Bh	INTEDG	11Bh	LCR	19Bh	EED
1Ch	Reserved	9Ch	CMPCON	11Ch	LSR	19Ch	EECON
1Dh	SELCON	9Dh	CMPCR	11Dh	THR/RBR	19Dh	EEPL
1Eh		9Eh	TBHP	11Eh	PWMDBx	19Eh	OPA0ADJ
1Fh	PWMxCON	9Fh	TBHD	11Fh	CCPxCON	19Fh	OPA1ADJ
20h General Purpose Register 80 Bytes 6Fh		A0h General Purpose Register 80 Bytes EFh		120h General Purpose Register 80 Bytes 16Fh		1A0h General Purpose Register 80 Bytes 1EFh	
70h General Purpose Register 16 Bytes (Common) 7Fh		F0h Mapped in Bank0 FFh		170h Mapped in Bank0 17Fh		1F0h Mapped in Bank0 1FFh	

Table 2-1 SFR Address Mapping

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial Value
00h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
01h	TMR0	TMR07	TMR06	TMR05	TMR04	TMR03	TMR02	TMR01	TMR00	xxxx xxxx
02h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
03h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
04h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	0xxx xxxx
05h	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
06h	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
07h	PORTC	-	-	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx
08h	-	-	-	-	-	-	-	-	-	-
09h	OSCCR	-	-	-	XSPD_STP	OPMD1	OPMD0	STPHOSC	SELHOSC	xxx0 0001
0Ah	PCHBUF				-	PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
0Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
0Ch	PIR1	INT2IF	INT1IF	WDTIF	-	EEIF	T5IF/ CCPIF	T4IF	T1IF	000x 0000
0Dh	PIR2	ADIF	LVDIF	CMPIF	-	SIMIF	LSRIF	TXIF	RXIF	000x 0010
0Eh	-	-	-	-	-	-	-	-	-	-
0Fh	TMRxL	TMRx7	TMRx6	TMRx5	TMRx4	TMRx3	TMRx2	TMRx1	TMRx0	xxxx xxxx
10h	TMRxH	-	-	-	-	-	-	TMRx9	TMRx8	xxxx xxxx
11h	TxCR1	-	-	TmxOE	VFSELx	TMx_HRC	TxOS	TxRL	TxEN	xx00 0000
12h	TxCR2	-	-	TxCS	TxCE	/PSxEN	PSxSEL2	PSxSEL1	PSxSEL0	xx11 1111
13h	BZ1CR	BZ1EN	-	-	-	BZ1FSEL3	BZ1FSEL2	BZ1FSEL1	BZ1FSEL0	0xxx1111
14h	AWUCON	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0	0000 0000
15h	BCDWUCON	WUPBCD7	WUPBCD6	WUPBCD5	WUPBCD4	WUPBCD3	WUPBCD2	WUPBCD1	WUPBCD0	0000 0000
16h	PORTACON30	PA3C1	PA3C0	PA2C1	PA2C0	PA1C1	PA1C0	PA0C1	PA0C0	0000 0000
17h	PORTACON74	PA7C1	PA7C0	PA6C1	PA6C0	PA5C1	PA5C0	PA4C1	PA4C0	0000 0000
18h	PORTBCON30	PB3C1	PB3C0	PB2C1	PB2C0	PB1C1	PB1C0	PB0C1	PB0C0	0000 0000
19h	PORTBCON74	PB7C1	PB7C0	PB6C1	PB6C0	PB5C1	PB5C0	PB4C1	PB4C0	0000 0000
1Ah	PORTCCON74	-	-	-	-	PC5C1	PC5C0	PC4C1	PC4C0	0000 0000
1Bh	PORTCCON30	PC3C1	PC3C0	PC2C1	PC2C0	PC1C1	PC1C0	PC0C1	PC0C0	0000 0000
1Ch										
1Dh	SELCON	CCPSEL1	CCPSLE0	HBMODE	BCDWU SEL1	BCDWU SEL0	TPSEL2	TPSEL1	TPSEL0	0000 0000
1Eh										
1Fh	PWMxCON							PWMxOEN	PWMxOAL	xxxx xx00

Table 2-1 Bank0 SFR bit Mapping (CONTINUED)

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial Value
80h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
81h	T0MD	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL2	PS0SEL1	PS0SEL0	0111 1111
82h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
83h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
84h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	0xxx xxxx
85h	IOSTA	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	1111 1111
86h	IOSTB	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	1111 1111
87h	IOSTC	-	-	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	xx11 1111
88h	-	-	-	-	-	-	-	-	-	-
89h	LVDCON	-	LVDOUT	-	-	LVDS3	LVDS2	LVDS1	LVDS0	xxxx 1111
8Ah	PCHBUF				-	PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
8Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
8Ch	PIE1	INT2IE	INT1IE	WDTIE	-	EEIE	T5IE/ CCPIE	T4IE	T1IE	000x 0000
8Dh	PIE2	ADIE	LVDIE	CMPIE	-	SIMIE	LSRIE	TXIE	RXIE	000x 0000
8Eh	-	-	-	-	-	-	-	-	-	-
8Fh	PCON	WDTEN	-	LVDEN	-	LVREN	-	-	-	1x0x 1xxx
90h	IRCR	IROSC 358M	-	-	-	-	IRCSEL	IRF57K	IREN	0xxx x000
91h	PWM1DUTYL	PWM1 DUTY7	PWM1 DUTY6	PWM1 DUTY5	PWM1 DUTY4	PWM1 DUTY3	PWM1 DUTY2	PWM1 DUTY1	PWM1 DUTY0	xxxx xxxx
92h	PWM1DUTYH	-	-	-	-	-	-	PWM1 DUTY9	PWM1 DUTY8	xxxx xxxx
93h	PWM2DUTYL	PWM2 DUTY7	PWM2 DUTY6	PWM2 DUTY5	PWM2 DUTY4	PWM2 DUTY3	PWM2 DUTY2	PWM2 DUTY1	PWM2 DUTY0	xxxx xxxx
94h	PWM2DUTYH	-	-	-	-	-	-	PWM2 DUTY9	PWM2 DUTY8	xxxx xxxx
95h	PWM3DUTYL	PWM3 DUTY7	PWM3 DUTY6	PWM3 DUTY5	PWM3 DUTY4	PWM3 DUTY3	PWM3 DUTY2	PWM3 DUTY1	PWM3 DUTY0	xxxx xxxx
96h	PWM3DUTYH	-	-	-	-	-	-	PWM3 DUTY9	PWM3 DUTY8	xxxx xxxx
97h	PWM4DUTYL	PWM4 DUTY7	PWM4 DUTY6	PWM4 DUTY5	PWM4 DUTY4	PWM4 DUTY3	PWM4 DUTY2	PWM4 DUTY1	PWM4 DUTY0	xxxx xxxx
98h	PWM4DUTYH	-	-	-	-	-	-	PWM4 DUTY9	PWM4 DUTY8	xxxx xxxx
99h	PWM5DUTYL	PWM5 DUTY7	PWM5 DUTY6	PWM5 DUTY5	PWM5 DUTY4	PWM5 DUTY3	PWM5 DUTY2	PWM5 DUTY1	PWM5 DUTY0	xxxx xxxx
9Ah	PWM5DUTYH	-	-	-	-	-	-	PWM5 DUTY9	PWM5 DUTY8	xxxx xxxx
9Bh	INTEDG	INT2EDG	EIS2	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0	0000 0101
9Ch	CMPCON	CMPEN	BIASEN	-	-	-	-	CMPOUT	CMPOE	00xx xxx0
9Dh	CMPCR	-	RBIAS_H	RBIAS_L	CMP_INV	PS1	PS0	NS1	NS0	x000 1100
9Eh	TBHP	-	-	-	TBHP4	TBHP3	TBHP2	TBHP1	TBHP0	xxxx xxxx
9Fh	TBHD	TBHD7	TBHD6	TBHD5	TBHD4	TBHD3	TBHD2	TBHD1	TBHD0	xxxx xxxx

Table 2-1 Bank1 SFR bit Mapping (CONTINUED)

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial Value
100h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
101h	TMR0	TMR07	TMR06	TMR05	TMR04	TMR03	TMR02	TMR01	TMR00	xxxx xxxx
102h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
103h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
104h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	0xxx xxxx
105h	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
106h	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
107h	PORTC	-	-	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx
108h	-	-	-	-	-	-	-	-	-	-
109h	ADMD	ADEN	START	GCHS	CHS4	CHS3	CHS2	CHS1	CHS0	0000 0000
10Ah	PCHBUF				-	PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
10Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
10Ch	ADDL	ADDL7	ADDL6	ADDL5	ADDL4	ADDL3	ADDL2	ADDL1	ADDL0	xxxx xxxx
10Dh	ADDH	ADDH7	ADDH6	ADDH5	ADDH4	ADDH3	ADDH2	ADDH1	ADDH0	xxxx xxxx
10Eh	ADCON1	EVHENB	-	EOC	ADFM	ADCK1	ADCK0	VHS1	VHS0	0x10 0011
10Fh	ADJMD	-	-	ADJ_SIGN	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0	xx00 0000
110h	PxCON	PxCON7	PxCON6	PxCON5	PxCON4	PxCON3	PxCON2	PxCON1	PxCON0	0000 0000
111h	ADCR	PxSEL1	PxSEL0	-	-	SHCK1	SHCK0	ADCR1	ADCR0	00xx 1010
112h	SIMDR	SIMD7	SIMD6	SIMD5	SIMD4	SIMD3	SIMD2	SIMD1	SIMD0	xxxx xxxx
113h	SIMCR	SPE	MEN	MSTA	SSB_PAD EN	RX_PAD EN	TX_PADEN	RCLK_PAD EN	UREN	0000 0000
114h	SPCR	SPIF	WCOL	-	MODF	CPOL	CKEG	SPR1	SPR0	00x0 0000
115h	MADR	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	-	0000 000x
116h	MFDR	-	-	-	FD4	FD3	FD2	FD1	FD0	xxx0 0000
117h	MCR	-	-	-	MTX	TXAK	-	-	-	xxx0 0xxx
118h	MSR	MCF	MAAS	MBB	MAL	-	SRW	MIF	RXAK	1000 x001
119h	DLL	DLL7	DLL6	DLL5	DLL4	DLL3	DLL2	DLL1	DLL0	0000 0000
11Ah	DLH	DLH7	DLH6	DLH5	DLH4	DLH3	DLH2	DLH1	DLH0	0000 0000
11Bh	LCR	LOOP	SBRK	PSTUCK	PEVEN	PREN	STPS	WL1	WL0	0000 0000
11Ch	LSR	-	TSRE	THRE	BRKINT	FERR	PERR	OVERR	READY	x110 0000
11Dh	THR/RBR	URD7	URD6	URD5	URD4	URD3	URD2	URD1	URD0	xxxx xxxx
11Eh	PWMDBx	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0000 0000
11Fh	CCPxCON	PWMxM1	PWMxM0	FBCH1	FBCH0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	0000 0000

Table 2-1 Bank2 SFR bit Mapping (CONTINUED)

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial Value
180h	INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0	xxxx xxxx
181h	T0MD	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL2	PS0SEL1	PS0SEL0	0111 1111
182h	PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	0000 0000
183h	STATUS	IRP	BK1	BK0	/TO	/PD	Z	DC	C	0001 1xxx
184h	FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0	0xxx xxxx
185h	IOSTA	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0	1111 1111
186h	IOSTB	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0	1111 1111
187h	IOSTC	-	-	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0	xx11 1111
188h	OPACON	OPA1EN	OPA1OEN	CMP1 MODE	OPA1_ ADC	OPA0EN	OPA0OEN	CMP0 MODE	OPA0_ ADC	0000 0000
189h	RFC	RFCEN				PSEL3	PSEL2	PSEL1	PSEL0	0xxx 0000
18Ah	PCHBUF	-	-	-		PCHBUF3	PCHBUF2	PCHBUF1	PCHBUF0	0000 0000
18Bh	INTCON	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF	0000 0000
18Ch	LCDCON0	LCDEN						LCDS1	LCDS0	0xxx xx00
18Dh	LCDCON1	P7COM	P6COM	P5COM	P4COM	P3COM	P2COM	P1COM	P0COM	0000 0000
18Eh	OSCCALL	OSCCAL7	OSCCAL6	OSCCAL5	OSCCAL4	OSCCAL3	OSCCAL2	OSCCAL1	OSCCAL0	xxxx xxxx
18Fh	OSCCALH	-	-	-	-	-	-	-	OSCCAL8	xxxx xxxx
190h	-	-	-	-	-	-	-	-	-	-
191h	-	-	-	-	-	-	-	-	-	-
192h	-	-	-	-	-	-	-	-	-	-
193h	-	-	-	-	-	-	-	-	-	-
194h	-	-	-	-	-	-	-	-	-	-
195h	-	-	-	-	-	-	-	-	-	-
196h	-	-	-	-	-	-	-	-	-	-
197h	-	-	-	-	-	-	-	-	-	-
198h	-	-	-	-	-	-	-	-	-	-
199h	-	-	-	-	-	-	-	-	-	-
19ah	EEA	EEA7	EEA6	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0	xxxx xxxx
19bh	EED	EED7	EED6	EED5	EED4	EED3	EED2	EED1	EED0	xxxx xxxx
19ch	EECON	TO1	TO0	-	EEW_ERR	EELOCK	-	WR	RD	00x0 1x00
19dh	EEPL	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	xxxx xxxx
19eh	OPA0ADJ	OPA0 DOUT	OPA0 COFM	OPA0CRS	OPA0 ADJ[4]	OPA0 ADJ[3]	OPA0 ADJ[2]	OPA0 ADJ[1]	OPA0 ADJ[0]	000x xxxx
19fh	OPA1ADJ	OPA1 DOUT	OPA1 COFM	OPA1CRS	OPA1 ADJ[4]	OPA1 ADJ[3]	OPA1 ADJ[2]	OPA1 ADJ[1]	OPA1 ADJ[0]	000x xxxx

Table 2-1 Bank3 SFR bit Mapping (CONTINUED)

2.2.2 Data Memory Register

2.2.2.1 INDF (Indirect Addressing Register)

The register INDF is not physically existed and it is used as indirect addressing mode. Any instruction accessing INDF actually accesses the register pointed by register IRP and FSR.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INDF	0x0	INDF[7:0]							
R/W Property		R/W							
Initial Value		xxxxxxxx							

2.2.2.2 FSR (Register File Selection Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR	0x4	FSR[7:0]							
R/W Property		R/W							
Initial Value		0	x	x	x	x	x	X	x

Bit 7 **FSR[7]:** Bank select bit 0 in indirect addressing mode

Bit 6:0 **FSR[6:0]:** Select one register out of 128 registers of specific Bank

2.2.2.3 TBHP (Table Access High Byte Address Pointer Register)

When instruction TABLEA is executed, the target address is constituted by TBHP[3:0] and ACC. ACC is the Low Byte of PC[11:0] and TBHP[3:0] is the high byte of PC[11:0].

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHP	0x9E	-	-	-	TBHP4	TBHP3	TBHP2	TBHP1	TBHP0
R/W Property		-	-	-	R/W	R/W	R/W	R/W	R/W
Initial Value		x	x	x	x	x	x	x	x

2.2.2.4 TBHD (Table Access High Byte Data Register)

When instruction TABLEA is executed, high byte of content of addressed ROM is loaded into TBHD[7:0] register. The Low Byte of content of addressed ROM is loaded to ACC.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHD	0x9F	TBHD7	TBHD6	TBHD5	TBHD4	TBHD3	TBHD2	TBHD1	TBHD0
R/W Property		R	R	R	R	R	R	R	R
Initial Value		x	x	x	x	x	x	x	x

2.2.3 Special Function Register

2.2.3.1 STATUS (Status Register)

The register STATUS contains result of arithmetic instructions and reasons to cause reset.

The STATUS register is shown in the following table and includes:

- The arithmetic status of the ALU.
- The reset status.
- The memory bank selection bits for the data memory.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	0x3	IRP	BK1	BK0	/TO	/PD	Z	DC	C
R/W Property		R/W	R/W	R/W	R/W*2	R/W*1	R/W	R/W	R/W
Initial Value		0	0	0	1	1	x	x	x

Bit 7 **IRP:** In indirect addressing mode, IRP & FSR[7] is used to select banks

IRP, FSR[7]=00, bank 0 is selected

IRP, FSR[7]=01, bank 1 is selected

IRP, FSR[7]=10, bank 2 is selected

IRP, FSR[7]=11, bank 3 is selected

Bit 6:5 **BK[1:0]:** In direct addressing mode, BK1 and BK0 is used to select banks

BK1, BK0=00, bank 0 is selected

BK1, BK0=01, bank 1 is selected

BK1, BK0=10, bank 2 is selected

BK1, BK0=11, bank 3 is selected

Bit 4 **/TO:** Time overflow flag bit

1 = After power-up or after instruction CLRWDT or SLEEP is executed

0 = WDT timeout is occurred

Bit 3 **/PD:** Time overflow flag bit

1 = After power-up or after instruction CLRWDT is executed

0 = After instruction SLEEP is executed

Bit 2 **Z:** Zero bit

1 = Result of logical operation is zero

0 = Result of logical operation is not zero

Bit 1 **DC:** Half Carry/half Borrow bit

1 = Carry from the 4th LSB is occurred for addition instruction or borrow from the 4th LSB is not occurred for subtraction instruction

0 = Carry from the 4th LSB is not occurred for addition instruction or borrow from the 4th LSB is occurred for subtraction instruction

Bit 0 **C:** Carry/Borrow bit

1 = Carry is occurred for addition instruction or borrow is not occurred for subtraction Instruction

0 = Carry is not occurred for addition instruction or borrow is occurred for subtraction Instruction

(*1) can be cleared by SLEEP instruction.

(*2) can be set by CLRWDT instruction.

2.2.3.2 SELCON (Select Register)

This register is used to control the mapping of virtual SFRs to physically existed SFRs so as to save the SFR space. As the following figure shows operation of virtual SFR and physical SFR: SFRx is a virtual register and SFR0 / SFR1 are physical registers. When SELCON, which can be CCPSEL, BCDWUSEL or TPSEL=0, read or write SFRx will actually access to SFR0. When SELCON=1, read or write SFRx will actually access to SFR1. Namely, which physical register is accessed depends on the current value of SELCON signal.

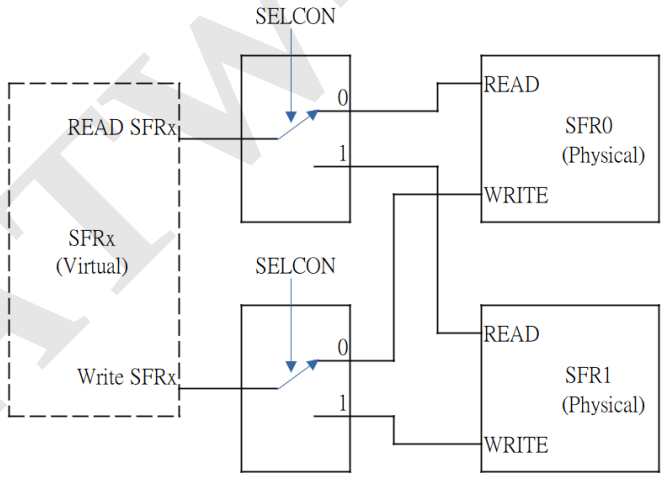


Figure 5 Operation of Virtual SFR

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SELCON	0x1D	CCPSEL[1:0]		HBMODE	BCDWUSEL[1:0]		TPSEL[2:0]		
R/W Property		R/W		R/W	R/W		R/W		
Initial Value(note*)		00		0	00		000		

Bit 7:6 **CCPSEL[1 :0] :** Virtual SFR CCPxCON and PWMDBx mapping

00 = CCPxCON and PWMDBx map to CCP1CON and PWMDB1 (default)

10 = CCPxCON and PWMDBx map to CCP2CON and PWMDB2

x1 = CCPxCON and PWMDBx map to CCP3CON and PWMDB3

Bit 5 **HBMODE :** CCP1~CCP3 driven mode

1 = [P1A,P1B]/[P2A,P2B]/[P3A,P3B] driven by PWM1/PWM2/PWM3

0 = [P1A,P1B]/[P2A,P2B]/[P3A,P3B] driven by PWM5/PWM4/PWM3

HB Output	HBMODE = 0		HBMODE = 1	
	Timer	PWM	Timer	PWM
P1A	5	5	1	1
P1B	5	5	1	1
P2A	4	4	1	2
P2B	4	4	1	2
P3A	1	3	1	3
P3B	1	3	1	3

Note: When SELCON is used by both the main code and the interrupt service routine, it may lead to program failure or incorrect readings. It is good practice to save the SELCON value to memory upon entering the interrupt service routine and restore it before returning from the interrupt

Bit 4:3 **BCDWUSEL:** Virtual SFR BCDWUCON mapping

00 = BCDWUCON map to BWUCON (default)

01 = BCDWUCON map to CWUCON

Bit 2:0 **TPSEL:** Virtual SFR TMRxL/TMRxH/TxCR1/TxCR2/PWMxCON mapping

001 = TMRxL / TMRxH / TxCR1 / TxCR2 / PWMxCON

map to TMR1L / TMR1H / T1CR1 /T1CR2 / PWM1CON (default)

010 = PWMxCON map to PWM2CON

011 = PWMxCON map to PWM3CON

100 = TMRxL / TMRxH / TxCR1 / TxCR2 / PWMxCON map to TMR4L / TMR4H / T4CR1 / T4CR2 / PWM4CON

101 = TMRxL / TMRxH / TxCR1 / TxCR2 / PWMxCON map to TMR5L / TMR5H / T5CR1 / T5CR2 / PWM5CON

SELCON	Virtual SFR	SELCON Value	Physical SFR
CCPSEL	CCPxCON	00	CCP1CON
		10	CCP2CON
		X1	CCP3CON
	PWMDBx	00	PWMDB1
		10	PWMDB2
		X1	PWMDB3
BCDWUSEL	BCDWUCON	00	BWUCON
		01	CWUCON
TPSEL	TMRxL	001	TRM1L
		100	TMR4L
		101	TMR5L
	TMRxH	001	TMR1H
		100	TMR4H
		101	TMR5H
	TxCR1	001	T1CR1
		100	T4CR1
		101	T5CR1
	TxCR2	001	T1CR2
		100	T4CR2
		101	T5CR2
	PWMxCON	001	PWM1CON
		010	PWM2CON
		011	PWM3CON
		100	PWM4CON
		101	PWM5CON

2.3 EEPROM Memory

Read and write access to EEPROM memory take place indirectly through 3 special function registers, namely, EEA, EED and EEPL. EEA register holds the EEPROM address to be access. EED register holds the data to be written, or the data read at the address in EEA. EEPL holds the unlock key to write EEPROM data. The following figure shows the block diagram how EEPROM operates.

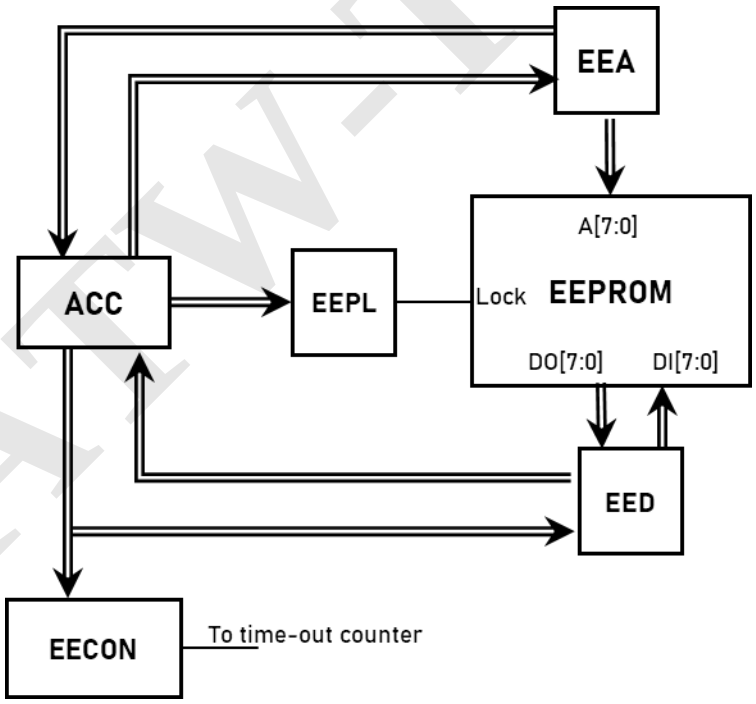


Figure 6 EEPROM Data Access

AT8F2481 provides EECON WR/RD bit to control the data, on the other hand, control the data path between EEPROM and EED register, according to the EEPROM address provided by EEA.

Before writing EEPROM data, 3 consecutive codes must be written to the EEPL register. These 3 code are C9H, 3AH and D3H. After these 3 codes are written to EEPL register, the EEPROM write protection is unlocked and data will be written to EEPROM by applying the write operation. The following are example code of EEPROM write unlock process.

The user should add a NOP instruction immediately after EEPROM write Mode instruction. (For example BSR EECON, 1)

```
; Write Serial Codes to Lock/Unlock EEPL
BCR     INTCON, GIE     ; Disable Global Interrupt
MOVIA   0xC9
MOVAR   EEPL
MOVIA   0x3A
MOVAR   EEPL
MOVIA   0xD3
MOVAR   EEPL
BSR     INTCON, GIE     ; Enable Global Interrupt

; Write EEPROM Data

; Method 1:
CLRWDT
BCR     INTCON, GIE     ; Disable Global Interrupt
MOVIA   0x45
MOVAR   EEA             ; Select EEPROM Address 0x45
MOVIA   0x23
MOVAR   EED             ; EEPROM Data 0x23
MOVIA   EE_TO_8ms | EE_WR
MOVAR   EECON           ; Write Data 0x23 to EEPROM Address 0x45
NOP
BSR     INTCON, GIE     ; Enable Global Interrupt

; Method 2:

CLRWDT
BCR     INTCON, GIE     ; Disable Global Interrupt
MOVIA   EE_TO_8ms
MOVAR   EECON
MOVIA   0x45
MOVAR   EEA             ; Select EEPROM Address 0x45
MOVIA   0x23
MOVAR   EED             ; EEPROM Data 0x23
BSR     EECON, EE_WR    ; Write Data 0x23 to EEPROM Address 0x45
NOP
BSR     INTCON, GIE     ; Enable Global Interrupt
```

```
; Read EEPROM Data
; Method 1:
BCR     INTCON, GIE     ; Disable Global Interrupt
MOVIA   0x45
MOVAR   EEA             ; Select EEPROM Address 0x45
MOVIA   EE_TO_8ms | EE_RD
MOVAR   EECON           ; Read Data of EEPROM Address 0x45
BSR     INTCON, GIE     ; Enable Global Interrupt

; Method 2:

BCR     INTCON, GIE     ; Disable Global Interrupt
MOVIA   EE_TO_8ms
MOVAR   EECON
MOVIA   0x45
MOVAR   EEA             ; Select EEPROM Address 0x45
BSR     EECON, EE_RD    ; Read Data of EEPROM Address 0x45
BSR     INTCON, GIE     ; Enable Global Interrupt
```

When writing operation is successfully executed or **EE write timeout or other interrupt end**, an EEIF interrupt will be launched if the EEIE is set to 1 and the global interrupt GIE is enabled.

Note: To prevent unwanted halt in a running program if EEPROM writes fail, enabling the watchdog reset or enabling EEPROM write time-out functions is suggested.

2.3.1 EEPROM Register

2.3.1.1 EEA (EEPROM Address Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEA	0x19A	EEA[7:0]							
R/W Property		R/W							
Initial Value		xxxxxxxx							

Bit 7:0 **EEA[7:0]:** Point to EEPROM address.

2.3.1.2 EED (EEPROM Data Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EED	0x19B	EED[7:0]							
R/W Property		R/W							
Initial Value		xxxxxxxx							

Bit 7:0 **EED[7:0]:** EEPROM data register.

2.3.1.3 EEPL (EEPROM Write Protect Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEPL	0x19D	PL[7:0]							
R/W Property		W							
Initial Value		xxxxxxxx							

Bit 7:0 **EEPL[7:0]:** EEPROM lock/unlock code.

2.3.1.4 EECON (EEPROM Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON	0x19C	TO1	TO0	-	EEW_ERR	EELOCK	-	WR	RD
R/W Property		R/W	R/W	-	R	R	-	W	W
Initial Value		0	0	x	0	1	x	0	0

Bit 7:6 **TO[1:0]:** EEPROM write time out period register.

The time-out period is shown in the following table:

TO[1:0]	Time-Out Period
00	8ms
01	8ms
10	16ms
11	32ms

Bit 5 **Unimplemented**

Bit 4 **EEW_ERR:** EEPROM write Time-out flag

1 = The EEPROM write is time-out (**Terminated prematurely due to another interrupt during the EEPROM write process)**

0 = The EEPROM write is not time-out

Bit 3 **EELOCK:** EEPROM write LOCK flag

1 = The EEPROM write is in lock state

0 = The EEPROM write is in unlock state

Bit 2 **Unimplemented**

Bit 1 **WR**: Write Control bit

1 = Initiates a write cycle

0 = Write cycle to the data EEPROM is complete

Bit 0 **RD**: Read Control bit

1 = Initiates an EEPROM read

0 = Does not initiate an EEPROM read

Note:

1. Before writing EEPROM, be sure to set initial data into EECON[7:6].

2. The minimum operating voltage of the EEPROM can be referenced in Section 25.6.

3. During the writing of the program EEPROM, the CPU will stop working, the CLRWDT command must be executed before the writing operation starts to avoid WDT overflow to reset the chip during this period.

4. When EECON[1:0]=11, will read only EEPROM.

2.4 Accumulator (ACC)

2.4.1 Overview

The ALU is an 8-bit wide Arithmetic Logic Unit that performs all arithmetic and logical operations in the MCU. It can carry out addition, subtraction, shifting, and logic operations on data. The ALU also controls the status flags (in the STATUS register), which reflect the result of operations.

The ACC register is an 8-bit register used to store the result of ALU operations. It is not part of the data memory but is located within the CPU, dedicated for use by the ALU during calculations. Therefore, it cannot be addressed directly and can only be accessed through specific instructions provided.

2.4.2 ACC Application

```
; Write Data Memory by Direct Addressing Mode
MOVR    0x20,ACC        ; Move GPR 0x20 Data to ACC
MOVAR   STATUS          ; Write ACC Data to STATUS
```

3. I/O Ports

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

3.1 Overview

AT8F2481 has 3 I/O ports: PORTA, PORTB and PORTC. PORTA is an 8-bit wide bi-directional port. The corresponding data direction register is IOSTA. PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is IOSTB. PORTC is a 6-bit wide bi-directional port. The corresponding data direction register is IOSTC.

When an I/O pin is configured as input pin, it may have Pull-High resistor or Pull-Low resistor. When an I/O pin is configured as output pin, it may be push-pull output or open-drain output. Each I/O pin has 2-bit SFR registers, that is PA0C[1:0] ~ PA7C[1:0], PB0C[1:0] ~ PB7C[1:0] and PC0C[1:0] ~ PC5C[1:0], to set these attributes. When I/O pin is in input mode, this 2-bit register is used to set pull-high or pull-low attribute. When I/O pin is in output mode, these 2-bit register is used to set push-pull or open-drain attribute, more detail descriptions are in the following table:

2-bit attribute SFR	Input	Output
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

Table 3-1 Summary of Pad I/O Feature

The sink current of each pin can be normal (26mA for V_{DD} =5V), large (38mA for V_{DD} =5V) or ultra (65 mA for VDD =5V) according to configuration words. When configured as output, the drive current of each pin can be normal (20mA for VDD =5V) or large (30mA for VDD =5V) according to configuration words. PA4~PA7 further provide ultra drive current (80mA for VDD =5V).

Check the following table for drive and sink current mode setting by configuration words:

Drive Mode	Sink Mode	Note
Normal	Normal	All Pin
Large	Large	All Pin
Large	Ultra	PA[3:0], PB[7:0], PC[5:0]
Ultra	Ultra	PA[7:4] Only

Table 3-2 Drive and Sink Current Mode

3.1.1 Block Diagram of IO Pins

OUT_EN: set pad attribute to output

WR_DATA: write data to pad.

RD_DATA: read pad.

RD_TYPE: select read pin or read latch.

PULLUP_ENB: enable 100KΩ Pull-High.

PULLDOWN_EN: enable Pull-Low.

OD_EN: set PAD as open-drain mode

WU_EN: enable pad change interrupt.

PADIF: pad change interrupt flag out.

ADCH_SEL: Enable pad to ADC channel input.

CMPCH_SEL: Enable pad to comparator channel input.

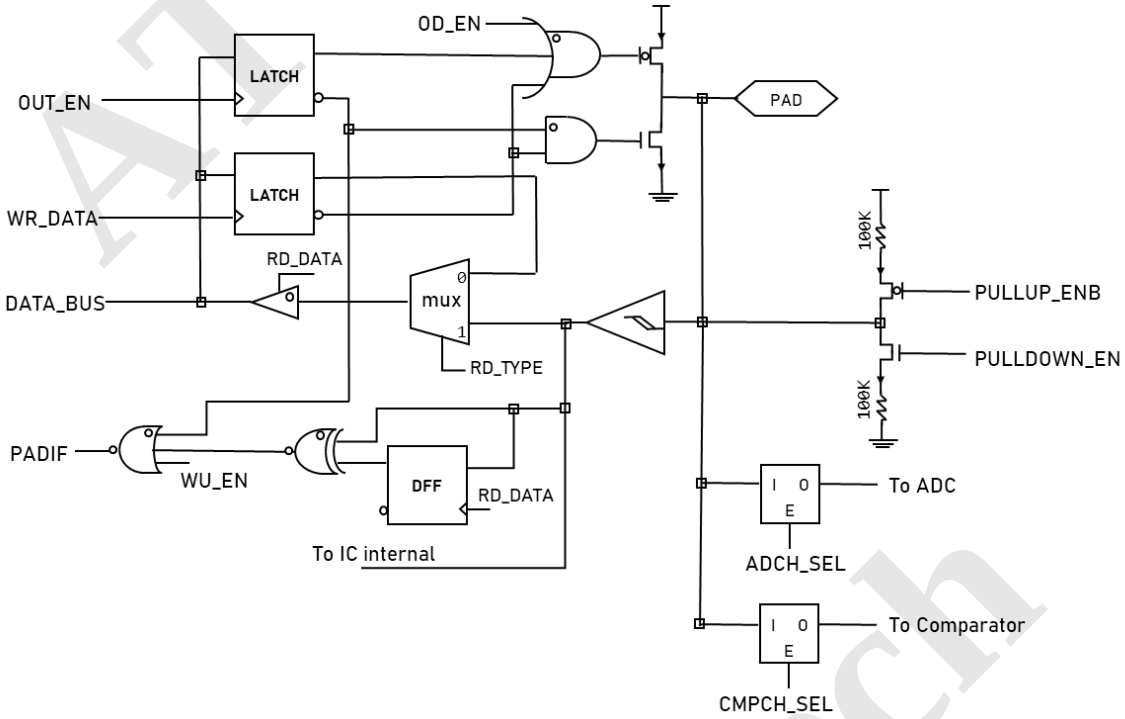


Figure 7 Block Diagram of PA0~PA4, PB3

OUT_EN: set pad attribute to output

WR_DATA: write data to pad.

RD_DATA: read pad.

RD_TYPE: select read pin or read latch.

PULLUP_ENB: enable 100KΩ Pull-High.

PULLDOWN_EN: enable Pull-Low.

OD_EN: set PAD as open-drain mode

WU_EN: enable pad change interrupt.

PADIF: pad change interrupt flag out.

ADCH_SEL: Enable pad to ADC channel input.

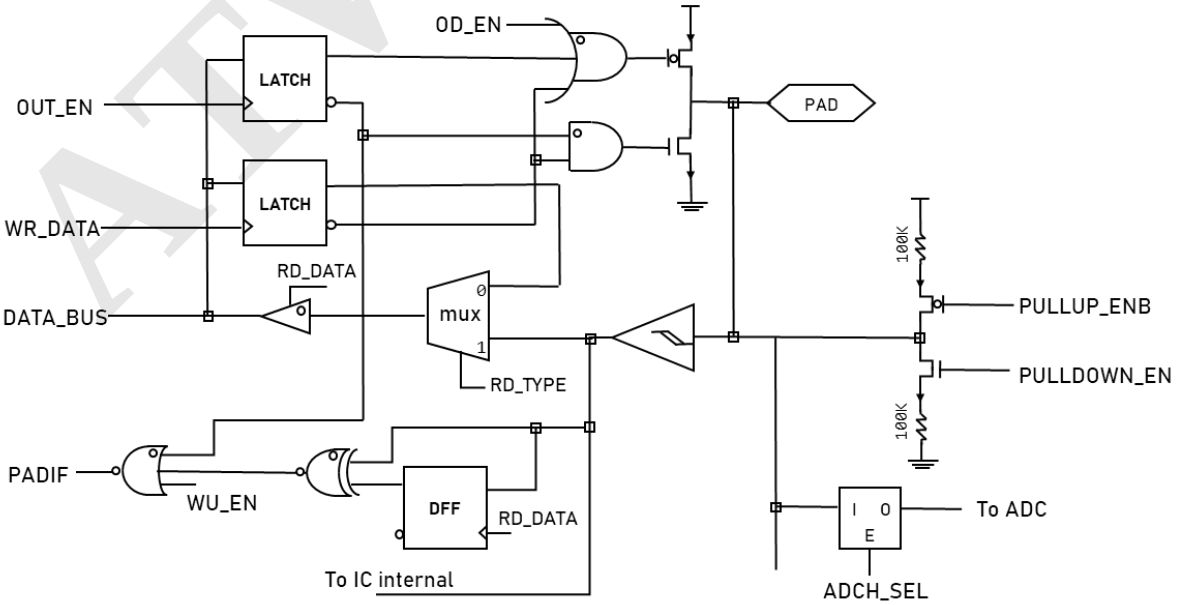


Figure 8 Block Diagram of PA5

OUT_EN: set pad attribute to output
WR_DATA: write data to pad.
RD_DATA: read pad.
RD_TYPE: select read pin or read latch.
PULLUP_ENB: enable 100KΩ Pull-High.
PULLDOWN_EN: enable Pull-Low.
OD_EN: set PAD as open-drain mode
WU_EN: enable pad change interrupt.
PADIF: pad change interrupt flag out.
ADCH_SEL: Enable pad to ADC channel input.
XTL_EN: Enable pad as XTAL pad.

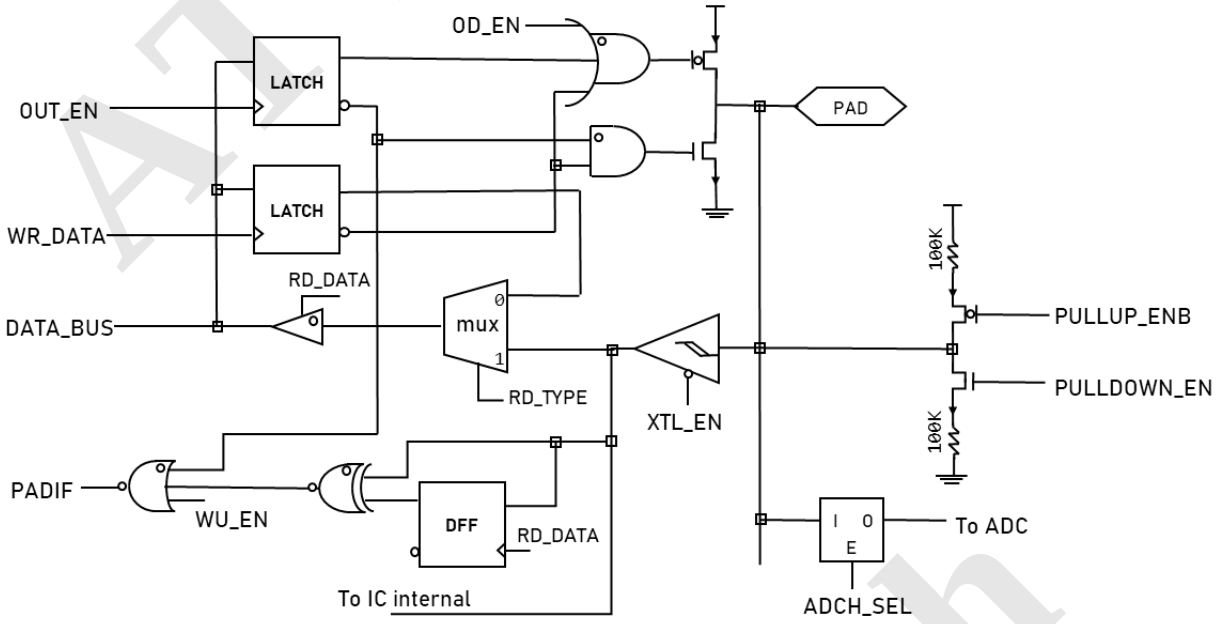


Figure 9 Block Diagram of PA6, PA7

OUT_EN: set pad attribute to output

WR_DATA: write data to pad.

RD_DATA: read pad.

RD_TYPE: select read pin or read latch.

PULLUP_ENB: enable 100KΩ Pull-High.

PULLDOWN_EN: enable Pull-Low.

OD_EN: set PAD as open-drain mode

WU_EN: enable pad change interrupt.

PADIF: pad change interrupt flag out.

ADCH_SEL: Enable pad to ADC channel input.

COM_EN: Enable PAD to output 1/2 VDD voltage.

VDD*0.5: 1/2 VDD voltage

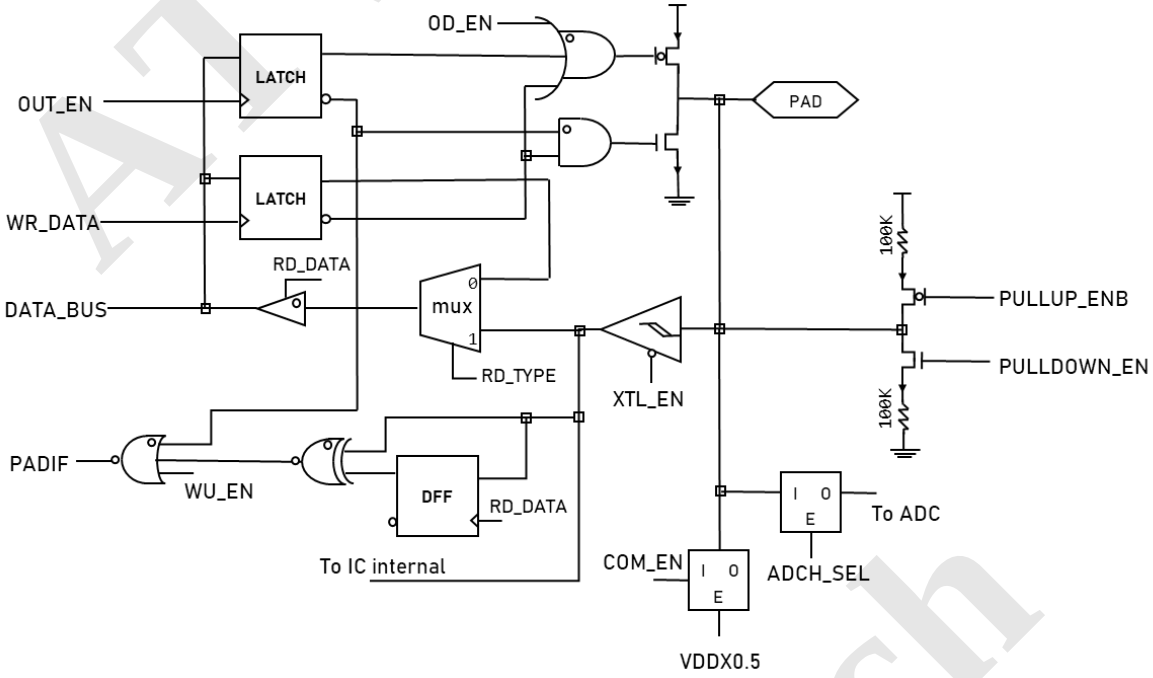


Figure 10 Block Diagram of PB4~PB7, PC2~PC5

OUT_EN: set pad attribute to output

WR_DATA: write data to pad.

RD_DATA: read pad.

RD_TYPE: select read pin or read latch.

PULLUP_ENB: enable 100KΩ Pull-High.

PULLDOWN_EN: enable Pull-Low.

OD_EN: set PAD as open-drain mode

WU_EN: enable pad change interrupt.

PADIF: pad change interrupt flag out.

ADCH_SEL: Enable pad to ADC channel input.

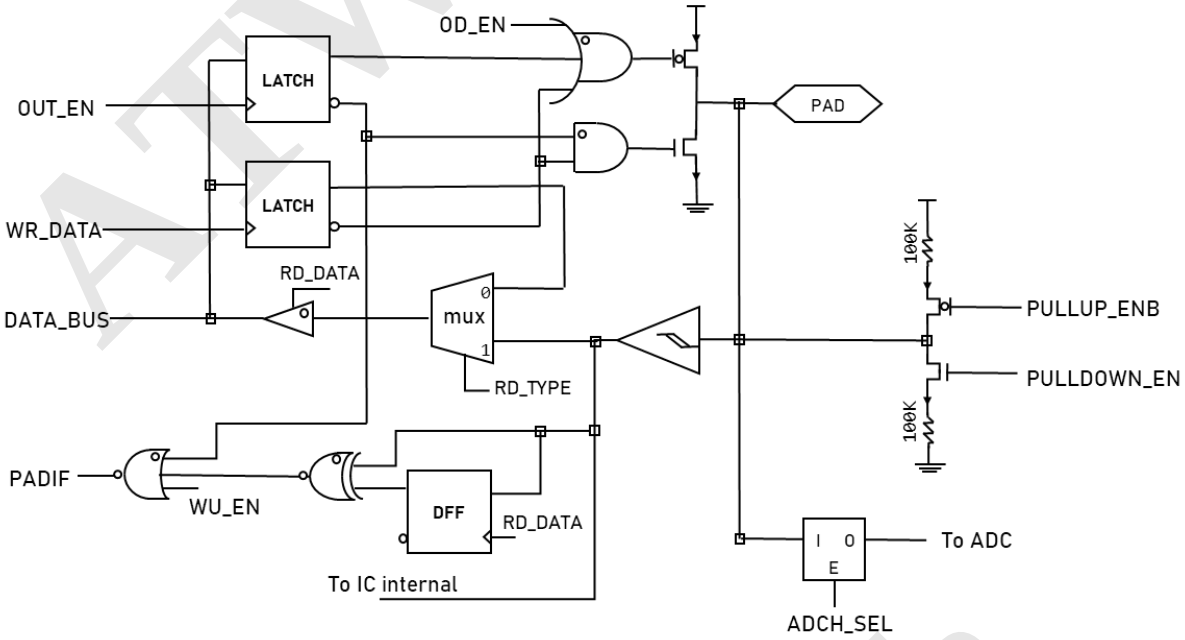


Figure 11 Block Diagram of PB0~PB2, PC0~1

3.2 I/O Ports Register

3.2.1 AWUCON (PORTA Wake-up Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AWUCON	0x14	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7:0 **WUPAx:** Enable/disable PAx wake-up function, $0 \leq x \leq 7$

1 = Enable PAx wake-up function.

0 = Disable PAx wake-up function.

3.2.2 BCDWUCON (PORTB/CWake-up Control Register)

Access the virtual SFR BCDWUCON is equivalent to access physical SFR BWUCON or CWUCON, according to BCDWUSEL[1:0] value.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BCDWUCON	0x15	WUPBCD7	WUPBCD6	WUPBCD5	WUPBCD4	WUPBCD3	WUPBCD2	WUPBCD1	WUPBCD0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7:0 **WUPBCDx:** Enable/disable PBx or PCx wake-up function
For BCDWUSEL[1:0]=00, 0 ≤ x ≤ 7
1 = Enable PBx wake-up function
0 = Disable PBx wake-up function
For BCDWUSEL[1:0]=01, 0 ≤ x ≤ 5
1 = Enable PCx wake-up function
0 = Disable PCx wake-up function

3.2.3 PORTACON30 / PORTACON74 / PORTBCON30 / PORTBCON74 / PORTCCON30 / PORTCCON74 (Port attribute Control Register)

These registers are used to set IO pad pull-up / pull-down attribute if corresponding pads are input. They also used to set IO pad push-pull or open-drain attribute if corresponding pads are output, see the following tables for details.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTACON30	0x16	PA3C[1:0]		PA2C[1:0]		PA1C[1:0]		PA0C[1:0]	
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTACON74	0x17	PA7C[1:0]		PA6C[1:0]		PA5C[1:0]		PA4C[1:0]	
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTBCON30	0x18	PB3C[1:0]		PB2C[1:0]		PB1C[1:0]		PB0C[1:0]	
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTBCON74	0x19	PB7C[1:0]		PB6C[1:0]		PB5C[1:0]		PB4C[1:0]	
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTCCON30	0x1B	PC3C[1:0]		PC2C[1:0]		PC1C[1:0]		PC0C[1:0]	
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTCCON74	0x1A	-		-		PC5C[1:0]		PC4C[1:0]	
R/W Property		-	-	-	-	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7:0 **PAxC[1:0]:** PORTA attribute setting, 0 ≤ x ≤ 7

PAxC[1:0]	(IOPAx=1) input	(IOPAx=0) output
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

PBxC[1:0]: PORTB attribute setting, 0 ≤ x ≤ 7

PBxC[1:0]	(IOPBx=1) input	(IOPBx=0) output
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

PCxC[1:0]: PORTC attribute setting, 0 ≤ x ≤ 5

PCxC[1:0]	(IOPCx=1) input	(IOPCx=0) output
00	Floating	Push-Pull
01	Pull-Down	Push-Pull
10	Pull-High	Open-Drain
11	-	Open-Drain

3.2.4 PORTA (PORTA Data Register)

While reading PORTA, it will get the status of the specific pin if that pin is configured as input pin. However, if that pin is configured as output pin, whether it will get the status of the pin or the value of the corresponding output data latch is depending on the configuration word “Read Output Data”. While writing to PORTA, data is written to PA’s output data latch.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA	0x5	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
R/W Property		R/W							
Initial Value		xxxxxxxx							

3.2.5 PORTB (PORTB Data Register)

While reading PORTB, it will get the status of the specific pin if that pin is configured as input pin. However, if that pin is configured as output pin, whether it will get the status of the pin or the value of the corresponding output data latch is depending on the configuration word “Read Output Data”. While writing to PORTB, data is written to PB’s output data latch.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTB	0x6	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W Property		R/W							
Initial Value		xxxxxxxx							

3.2.6 PORTC (PORTC Data Register)

While reading PORTC, it will get the status of the specific pin if that pin is configured as input pin. However, if that pin is configured as output pin, whether it will get the status of the pin or the value of the corresponding output data latch is depending on the configuration option “Read Output Data”. While writing to PORTC, data is written to PC’s output data latch.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTC	0x7	-	-	PC5	PC4	PC3	PC2	PC1	PC0
R/W Property		-	-	R/W					
Initial Value		xxxxxxxx							

3.2.7 IOSTA (PORTA I/O Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTA	0x85	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		1	1	1	1	1	1	1	1

Bit 7:0 **IOPAx:** PAx I/O mode selection, $0 \leq x \leq 7$

1 = PAx is input mode

0 = PAx is output mode

3.2.8 IOSTB (PORTB I/O Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTB	0x86	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		1	1	1	1	1	1	1	1

Bit 7:0 **IOPBx:** PBx I/O mode selection, $0 \leq x \leq 7$

1 = PBx is input mode

0 = PBx is output mode

3.2.9 IOSTC (PORTC I/O Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTC	0x87	-	-	IOPC5	IOPC4	IOPC3	IOPC2	IOPC1	IOPC0
R/W Property		-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		x	x	1	1	1	1	1	1

Bit 7:6 **Unimplemented**

Bit 5:0 **IOPCx:** PCx I/O mode selection, $0 \leq x \leq 5$

1 = PCx is input mode

0 = PCx is output mode

3.2.10 PxCON (Port Analog Pin Control Register)

Access the virtual SFR PxCON is equivalent to access physical SFR PACON, PBCON or PCCON, according to PxSEL[1:0] value.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxCON	0x110	PxCON7	PxCON6	PxCON5	PxCON4	PxCON3	PxCON2	PxCON1	PxCON0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

When PxSEL=00, PxCON=PACON,

Bit 7:0 **PACON[7:0]:** PORTA analog pin select

1 = PAi is pure analog pin, input buffer is disabled for power saving, $0 \leq i \leq 7$

0 = PAi can be analog or digital pin, $0 \leq i \leq 7$

When PxSEL=01, PxCON=PBCON,

Bit 7:0 **PBCON[7:0]:** PORTB analog pin select

1 = PBi is pure analog pin, input buffer is disabled for power saving, $0 \leq i \leq 7$

0 = PBi can be analog or digital pin, $0 \leq i \leq 7$

When PxSEL=10, PxCON=PCCON,

Bit 5:0 **PCCON[5:0]:** PORTC analog pin select

1 = PCi is pure analog pin, input buffer is disabled for power saving, $0 \leq i \leq 5$

0 = PCi can be analog or digital pin, $0 \leq i \leq 5$

4. Timer0

4.1 Overview

Timer0 is an 8-bit up-count timer and its operation is enabled by register bit T0EN (T0MD[6]). Writing to Timer0 will set its initial value. Reading from Timer0 will show its current count value.

The clock source to Timer0 can be from instruction clock, external pin EX_CKI0 or low speed clock Low Oscillator Frequency according to register bit T0CS and LCKTM0 (T0MD[5] and T0MD[7]). When T0CS is 0, instruction clock is selected as Timer0 clock source. When T0CS is 1 and LCKTM0 is 0, EX_CKI0 is selected as Timer0 clock source. When T0CS is 1 and LCKTM0 is 1 (and Timer0 source must set to 1), Low Oscillator Frequency (I_LRC or E_LXT, depends on configuration word) output is selected. Summarized table is shown below.

Timer0 clock source	T0CS	LCKTM0	Timer0 source	Low Oscillator Frequency
Instruction clock	0	X	X	X
EX_CKI0	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

Table 4-1 Summary of Timer0 Clock Source Control

Moreover, the active edge of EX_CKI0 or Low Oscillator Frequency to increase Timer0 can be selected by register bit T0CE (T0MD[4]). When T0CE is 1, high-to-low transition on EX_CKI0 or Low Oscillator Frequency will increase Timer0. When T0CE is 0, low-to-high transition on EX_CKI0 or Low Oscillator Frequency will increase Timer0. When using Low Oscillator Frequency as Timer0 clock source, it is suggested to use prescaler0 (see below descriptions) and the ratio set to more than 4, or missing count may happen. The prescaler0 frequency must be at least **two times slower** than the $F_{INST.}$ frequency.

Before Timer0 clock source is supplied to Timer0, it can be divided by Prescaler0 if register bit PS0WDT (T0MD[3]) is clear to 0. When writing 0 to PS0WDT by instruction, Prescaler0 is assigned to Timer0 and Prescaler0 will be clear after this instruction is executed. The dividing rate of Prescaler0 is determined by register bits PS0SEL[2:0] which is from 1:2 to 1:256.

When Timer0 is overflow, the register bit T0IF (INTCON[2]) will be set to 1 to indicate Timer0 overflow event is occurred. If register bit T0IE (INTCON[5]) and GIE are both set to 1, interrupt request will occur and interrupt service routine will be executed. T0IF will not be clear until firmware writes 0 to T0IF.

The block diagram of Timer0 and WDT is shown in the figure below.

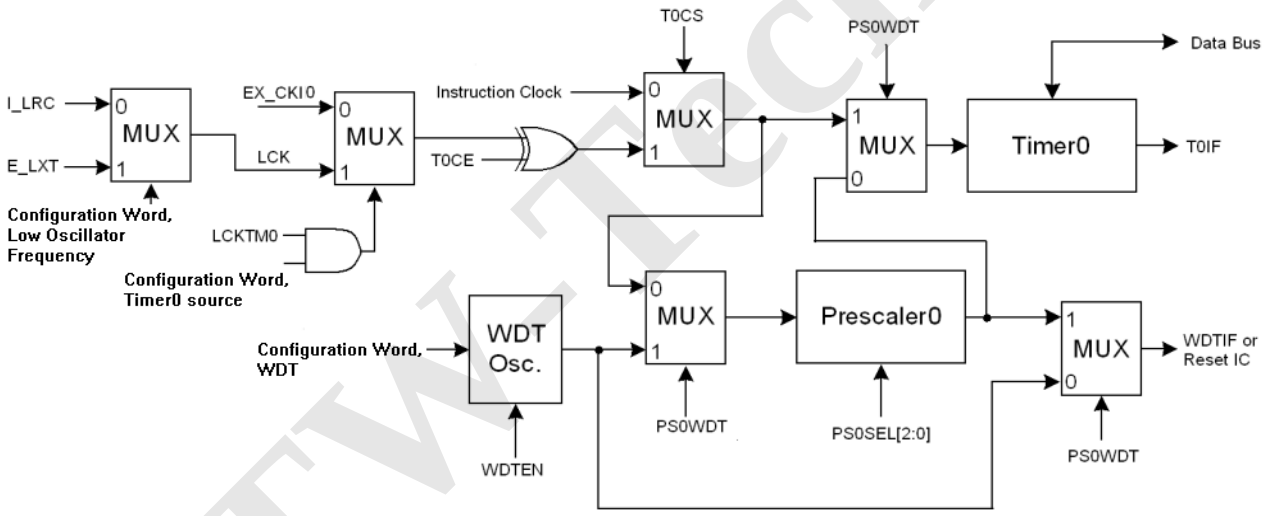


Figure 12 Block Diagram of Timer0 and WDT

4.2 Timer0 Control Register

4.2.1 T0MD Register

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0MD	0x81	LCKTM0	T0EN	T0CS	T0CE	PS0WDT	PS0SEL[2:0]		
R/W Property		R/W							
Initial Value(note*)		0	1	1	1	1	111		

When T0CS=0, Instruction clock F_{INST} is selected as Timer0 clock source.

Bit 7 **LCKTM0:** Timer0 clock source select

When T0CS=1, timer 0 clock source can be optionally selected to be low frequency oscillator.

1 = Low Oscillator Frequency (I_LRC or E_LXT, depends on configuration word Low Oscillator Frequency) output replaces pin EX_CKI0 as Timer0 clock source.

0 = external clock on pin EX_CKI0 is selected as Timer0 clock source.

Note: For more detail descriptions of Timer0 clock source select, please see Timer0 section.

Bit 6 **T0EN:** Enable/disable Timer0

1 = Enable Timer0

0 = Disable Timer0

Bit 5 **T0CS:** Timer0 clock source selection

1 = External clock on pin EX_CKI0 or Low Oscillator Frequency (I_LRC or E_LXT) is selected

0 = Instruction clock F_{INST} is selected

Bit 4 **T0CE:** Timer0 external clock edge selection

1 = Timer0 will increase one while high-to-low transition occurs on pin EX_CKI0

0 = Timer0 will increase one while low-to-high transition occurs on pin EX_CKI0

Note: T0CE is also applied to Low Oscillator Frequency as Timer0 clock source condition.

Bit 3 **PS0WDT:** Prescaler0 assignment

1 = Prescaler0 is assigned to WDT

0 = Prescaler0 is assigned to Timer0

Note: Always set PS0WDT and PS0SEL[2:0] before enabling watchdog or timer0 interrupt, or reset or interrupt may be falsely triggered.

Bit 2:0 **PS0SEL[2:0]:** Prescaler0 dividing rate selection. The rate depends on Prescaler0 is assigned to Timer0 or WDT. When Prescaler0 is assigned to WDT, the dividing rate is dependent on which timeout mechanism is selected.

PS0SEL[2:0]	Dividing Rate		
	PS0WDT=0 (Timer0)	PS0WDT=1 (WDT Reset)	PS0WDT=1 (WDT Interrupt)
000	1:2	1:1	1:2
001	1:4	1:2	1:4
010	1:8	1:4	1:8
011	1:16	1:8	1:16
100	1:32	1:16	1:32
101	1:64	1:32	1:64
110	1:128	1:64	1:128
111	1:256	1:128	1:256

4.2.2 TMR0 (Timer0 Register)

When read the register TMR0, it actually read the current running value of Timer0. Write the register TMR0 will change the current value of Timer0. Timer0 clock source can be from instruction clock FINST, or from external pin EX_CKI0, or from Low Oscillator Frequency according to OPTION register and configuration word setting.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0	0x1	TMR0[7:0]							
R/W Property		R/W							
Initial Value		xxxxxxxx							

5. Timer1 / Timer4 / Timer5

5.1 Overview

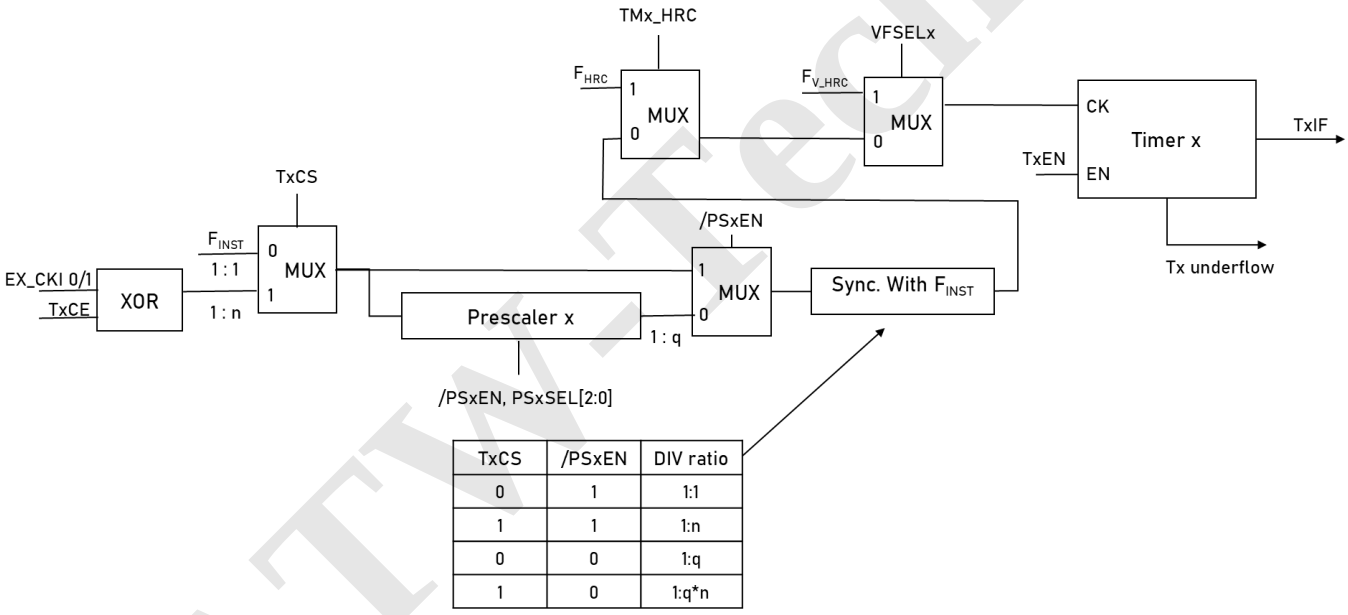
Timer1, Timer4 and Timer5 have similar structure. They all are 10-bit down-count timer with Prescaler whose dividing ratio is programmable. But they also different in other specs. The Timer1, Timer4 and Timer function comparison is listed in the following tables.

Timer clock source	Timer1	Timer4	Timer5	Note
Counter	10-bit			
Up or Down count	Down count			
Count mode	One-shot and Non-Stop			
Reload mode	Reload 0x3ff or register			
Prescaler dividing rate	8 selectable			
High oscillator clock input	YES			
External clock	EX_CKI0	EX_CKI1	EX_CKI1	Rising / Falling edge
PWM (for timer frame)	PWM1/2/3	PWM4	PWM5	
Timer toggle output	Y	N	N	
Buzzer (frequency source)	Y	N	N	
CCP1 compare timer	N	Low byte	High byte	
CCP1 capture timer	N	Low byte	High byte	
CCP1 Full-bridge	N	N	Y	
CCP1 Half-bridge	N	N	Y	Dead-band control
CCP2 Half-bridge	N	Y	N	
CCP3 Half-bridge	Y	N	N	
RFC	Y	N	N	

Table 5-1 Summary of Timer0 Clock Source Control

The Timer1/4/5 operation are controlled by virtual SFR TMRxL / TMRxH / TxCR1 / TxCR2 and SELCON(TPSEL[2:0]). TPSEL[2:0] is used to control the mapping of these virtual SFR to physically exist timer SFR. For example, when TPSEL[2:0]=4, the virtual SFR TMRxL / TMRxH / TxCR1 / TxCR2 will mapping to SFR TMR4L / TMR4H / T4CR1 / T4CR2. It means access to these virtual SFR will actually access to timer4 SFR. For AT8F2481, TPSEL can be 1, 4 or 5, which corresponding to Time1, Time4 and Timer5.

Note: Throughout this section, generic reference is used for register and bit names that are the same, except for an 'x' variable that indicates the item's association to a specific Timer module. For the sake of clarity, all module operations are described generically, and are equally applicable to all Timer modules. The difference between timers will be pointed out if necessary.



TxCS	/PSxEN	DIV ratio
0	1	1:1
1	1	1:n
0	0	1:q
1	0	1:q*n

Figure 13 Block Diagram of Timer1 / 4 / 5

Configuration Option	High IRC Frequency	F_{HRC}
7	24MHz	24MHz
6	20MHz	20MHz
5	16MHz	16MHz
4	8MHz	16MHz
3	4MHz	16MHz
2	2MHz	16MHz
1	1MHz	16MHz

Table 5-2

About F_{HRC} frequency, please refer table 5-2,

F_{V_HRC} frequency can be choose as is 32MHz, 20.8MHz, 16MHz, or 13.6MHz, which is decided by configuration word.

The timerx clock source can be selected by VFSELx(TxCR1[4]). When VFSELx=1, not matter what TMx_HRC=1 or TMx_HRC=0, the timerx clock source is direct V_HRC output. The timerx clock source can be selected by TMx_HRC(TxCR1[3]). When TMx_HRC=1 and VFSELx=0, the timerx clock source is direct I_HRC output. When TMx_HRC=0 and VFSELx=0, the timerx clock source can be instruction clock (F_{INST}) or external clock (EX_CKI0 or EX_CKI1) selected by TxCS(TxCR2[5]).

The timerx clock source division ratio is selected by TxCS and Prescalerx, as the above figure shows. When TxCS=0, the base division ratio is 1. When TxCS=1, the base division ratio is external clock (EX_CKI0 or EX_CKI1) cycle divided by CPU cycle (F_{INST}). When /PSxEN=0, Prescalerx is enabled, the prescaler dividing rate is from 1:2 to 1:256 determined by register bits PS1SEL[2:0] (T1CR2[2:0]). When /PSxEN=1, Prescalerx is disabled, the prescaler dividing rate is 1:1. The final timerx clock source division ratio is the prescaler division ratio multiply by the base division ratio. Note that due to the frequency of HRC clock (F_{HRC}) is higher

than the frequency of CPU clock (F_{INST}), when timerx clock source is HRC clock output (F_{HRC}), the final timerx clock source division ratio must be 1, that is, set TxCS=0 and /PSxEN=1.

Note the timer external clock input is different for Timer1 and Timer4/Timer5. Timer1's external clock is EX_CKI0 (PA4), timer4 and timer5's external clock is EX_CKI1 (PA1 or PA2 by option). When EX_CKI0 or EX_CKI1 is selected as clock source for timer, the active edge to decrease Timerx is determined by register bit TxCE (TxCR2[4]). When TxCE is 1, high-to-low transition on EX_CKI will decrease Timer. When TxCE is 0, low-to-high transition on EX_CKI will decrease Timer.

The operation of Timer can be enabled or disabled by register bit TxEN (TxCR1[0]).

The MSB 2-bit of the 10-bit timers is from TMRxH[1:0], whereas the LSB 8-bit is from TMRxL[7:0].

When reading register TMRxH, it will obtain current MSB value of 10-bit down-count Timerx, that is TMRx[9:8]. When reading register TMRxL, it will obtain current LSB value of 10-bit down-count Timerx, that is TMRx[7:0]. When writing register TMRxH, it will write data to TMRx[9:8] reload register When writing register TMRxL, it will write data to TMRx[7:0] reload register. When user write Timerx reload register, write TMRxH first and then write TMRxL.

Timer provides two kinds of operating mode: one is One-Shot mode and the other is Non-Stop mode. When register bit TxOS (TxCR1[2]) is 1, One-Shot mode is selected. Timer will count down once from initial value stored on register TMRx[9:0] to 0x00, i.e. underflow is occurred. When register bit TxOS(TxCR1[2]) is 0, Non-Stop mode is selected. When underflow is occurred, there are two selections to start next down-count which is determined by register bit TxRL (TxCR1[1]). When TxRL is 1, timer is in auto-reload mode, the initial value stored on timer reload register will be restored and start next down-count from this initial value. When TxRL is 0 (continuous mode), Timerx will start next down-count from 0x3FF.

Timerx has adopted double buffer mechanism in auto-reload mode (TxOS=0 and TxRL=1) : When TxEN=1, Timerx reload register will not be uploaded to Timer1 counter(TMRx[9:0]) until Timerx overflow occurs. However, when TxEN=0, Timerx reload register will be uploaded to Timerx counter(TMRx[9:0]) after writing TMRxL immediately, no matter timerx is in one-shot mode, auto-reload mode or continuous-mode.

When Timerx is underflow, the register bit TxIF (PIR[2:0] for T5IF, T4IF and T1IF respectively will be set to 1, indicate Timerx underflow event is occurred. If register bit TxIE (PIE[2:0] for T5IE, T4IE and T1IE respectively) and GIE are both set to 1, interrupt request will occur and interrupt service routine will be executed. TxIF will not be clear until firmware writes 0 to TxIF.

The timing chart of Timer is shown in the following figure.

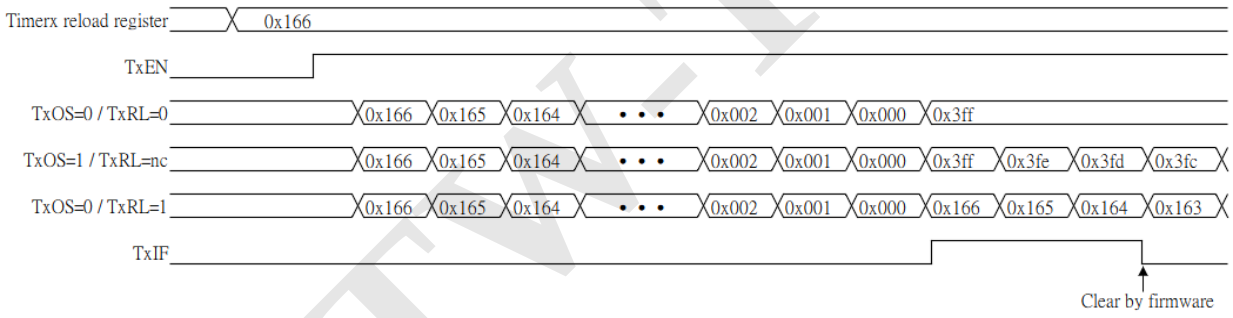


Figure 14 Timer1 / 4 / 5 Timing Chart

5.2 Timer1 / Timer4 / Timer5 Control Register

5.2.1 TMRxL (Timerx low byte Register)

Access the virtual SFR TMRxL is equivalent to access physical SFR TMR1L, TMR4L or TMR5L, according to SELCON(TPSEL[2:0]) value.

When reading register TMRxL and TPSEL[2:0]=001, it will obtain current LSB value of 10-bit down-count Timer1, that is TMR1[7:0]. When writing register TMRxL and TPSEL[2:0]=001, it will write data to TMR1L reload register, and if T1EN=0, it will also write to TMR1[7:0] current content. The operations hold for TPSEL[2:0] = 100 (Timer4) or 101 (Timer5) conditions.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRxL	0x0F	TMRx[7:0]							
R/W Property		R/W							
Initial Value		xxxxxxxx							

5.2.2 TMRxH (Timer High Byte Register)

Access the virtual SFR TMRxH is equivalent to access physical SFR TMR1H, TMR4H or TMR5H, according to SELCON(TPSEL[2:0]) value.

When reading register TMRxH and TPSEL[2:0]=001, it will obtain current MSB value of 10-bit down-count Timer1, that is TMR1[9:8]. When writing register TMRxL and TPSEL[2:0]=001, it will write data from TMR1H reload register to TMR1[9:8] current content if T1EN=0. When writing register TMRxH and TPSEL[2:0]=001, it will write data to TMR1H reload register. The operations also hold for TPSEL[2:0] = 100 (Timer4) or 101 (Timer5) conditions.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRxH	0x10	-	-	-	-	-	-	TMRx[9:8]	
R/W Property		-	-	-	-	-	-	R/W	
Initial Value		x	x	x	x	x	x	xx	

5.2.3 TxCR1 (Timer Control Register1)

Access the virtual SFR TxCR1 is equivalent to access physical SFR T1CR1, T4CR1 or T5CR1, according to TPSEL[2:0] value.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TxCR1	0x11	-	-	TMxOE	VFSELx	TMx_HRC	TxOS	TxRL	TxEN
R/W Property		-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		x	x	0	0	0	0	0	0

This register is used to configure Timerx functionality. (x=1, 4 or 5)

Bit 7:6 **Unimplemented**

Bit 5 **TMxOE:** Enable / Disable Timerx match output, TxOUT toggle output when Timerx underflow occurs. **(*1), (*2)**

1 = Enable TxOUT output to pad PB5

0 = PB5 is GPIO

Bit 4 **VFSELx:** Timerx special clock source selection **(*3)**

1 = Timerx & PWM clock source is special High Oscillator clock

0 = Timerx & PWM clock source selection depends on T1CS register bit

Bit 3 **TMx_HRC:** Timerx clock source selection

1 = Timerx and related PWM clock source is High Oscillator clock

0 = Timerx and related PWM clock source selection depends on T1CS register bit

Bit 2 **TxOS:** Configure Timerx operating mode while underflow is reached

1 = One-Shot mode. Timerx will count once from the initial value to 0x00

0 = Non-Stop mode. Timerx will keep down-count after underflow

TxOS	TxRL	Timerx Down-Count Functionality
0	0	Timerx will count from reload value down to 0x00. When underflow is reached, 0x3FF is reloaded and continues down-count.
0	1	Timerx will count from reload value down to 0x00. When underflow is reached, reload value is reloaded and continues to down-count.
1	x	Timerx will count from initial value down to 0x00. When underflow is reached, Timerx will stop down-count.

Bit 1 **TxRL:** Configure Timerx down-count mechanism while Non-Stop mode is selected (TxOS=0)

1 = Initial value is reloaded from reload register

0 = Continuous down-count from 0x3FF when underflow is occurred

Bit 0 **TxEN:** Enable/disable Timerx

1 = Enable Timerx

0 = Disable Timerx

Note:

1. For AT8F2481, only Timer1 has Timer match output T1OUT.

2. If both T1OUT and PWM1 are configured to output to PB5, T1OUT has higher priority than PWM1 output.

3. VFSELx has higher priority than TMx_HRC.

5.2.4 TxCR2 (Timer Control Register2)

Access the virtual SFR TxCR2 is equivalent to access physical SFR T1CR2, T4CR2 or T5CR2, according to TPSEL[2:0] value.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TxCR2	0x12	-	-	TxCS	TxCE	/PSxEN	PSxSEL[2:0]		
R/W Property		-	-	R/W	R/W	R/W	R/W		
Initial Value		x	x	1	1	1	111		

This register is used to configure Timerx functionality. (x=1, 4 or 5)

Bit 7:6 **Unimplemented**

Bit 5 **TxCS:** Timerx clock source selection

1 = External clock on pin EX_CKIx is selected

0 = Instruction clock is selected

Bit 4 **TxCE:** Timerx clock source selection

1 = Timerx will decrease one while high-to-low transition occurs on external clock pin EX_CKIx

0 = Timerx will decrease one while low-to-high transition occurs on external clock pin EX_CKIx

Timer	EXCKIx	Note
0	PA4	
1	PA4	
4	PA1	External clock option=0
	PA2	External clock option=1
5	PA1	External clock option=0
	PA2	External clock option=1

Bit 3 **/PSxEN:** Timerx clock source selection

1 = Disable Timerx Prescaler

0 = Enable Timerx Prescaler

Bit 2:0 **PSxSEL[2:0]:** Timerx Prescaler Dividing Rate

PSxSEL[2:0]	Dividing Rate
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Note: Always set PSxSEL[2:0] at PSxEN=1, or interrupt may be falsely triggered.

6. PWM

AT8F2481 has the ability to output 5 PWM waveforms with 10-bit resolution at the same time. PWM1, PWM2 and PWM3 share the same time frame (timer1), whereas PWM4 and PWM5 each has independent time frame (timer4 and timer5). The PWM output pads are determined by configuration words, as the following table shows.

Configuration Word	1	2	3	4	5
PWM1_PAD	PB5	PB1	PA3	PA5	
PWM2_PAD	PB4	PA4	PB7	PA7	PB3
PWM3_PAD	PA6	PA2	PB6	PB0	
PWM4_PAD	PC3	PB3	PA1	PC1	PA4
PWM5_PAD	PB2	PC0	PC5	PA0	PC2

Table 6-1 PWM Output PAD Configuration

6.1 PWM1

6.1.1 Overview

The PWM1 output can be available on one of the I/O pin PB5, PB1, PA3 or PA5 (selected by config. options). PWM1 is enabled by setting SELCON(TPSEL[2:0]=001) and PWMxOEN=1 sequentially. PWM1 will become output pin automatically. The active state of PWM1 output is determined by register bit PWMxOAL. When setting PWMxOAL is 1 at TPSEL[2:0]=001, PWM1 output is active low. When PWMxOAL is 0 at TPSEL[2:0]=001, PWM1 output is active high. Moreover, the duty cycle and frame rate of PWM1 are both programmable. The duty cycle is determined by registers PWM1DUTY[9:0] (PWM1DUTYH[1:0] and PWM1DUTYL[7:0]). When PWM1DUTY is 0, PWM1 output will be never active. When PWM1DUTY is 0x3FF, PWM1 output will be active for 1023 Timer1 input clocks. The frame rate is determined by Timer1 period. PWM1DUTY value must be less than or equal to Timer1 period.

The update of AT8F2481 PWM DUTY adopts double buffer mechanism when PWM1 is enabled (PWMxOEN=1): user write PWM1DUTYL and PWM1DUTYH will not take effect until timer1 overflow. It means new PWM1 Duty cycle can only be available on the next timer1 period.

When PWM1 is disabled, write PWM1DUTY[9:8] MSB 2 bits(PWM1DUTYH[1:0]) first, then write PWM1DUTYL[7:0], this ensure the first timer1 period gets the right PWM DUTY

The block diagram of PWM1 is illustrated in the following figure.

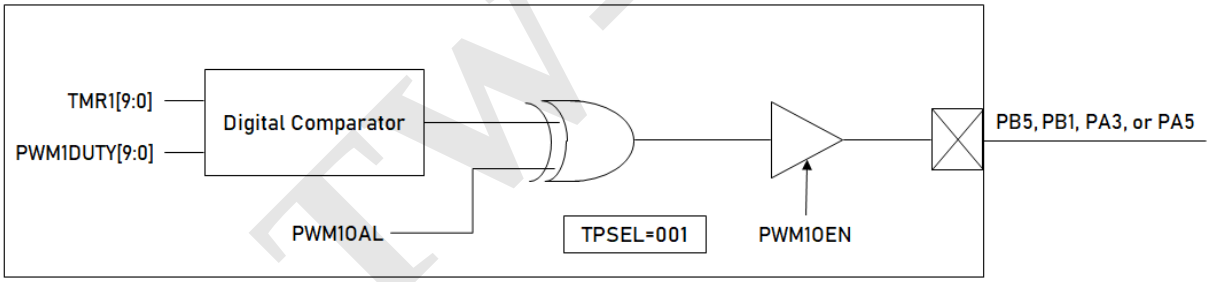


Figure 15 PWM1 Block Diagram

6.2 PWM2

6.2.1 Overview

The PWM2 output can be available on one of the I/O pin PB4, PA4, PB3, PB7 or PA7 (selected by configuration word). PWM2 is enabled by setting SELCON(TPSEL[2:0]=010) and PWMxOEN=1 sequentially. PWM2 will become output pin automatically. The active state of PWM2 output is determined by register bit PWMxOAL. When setting PWMxOAL is 1 at TPSEL[2:0]=010, PWM2 output is active low. When PWMxOAL is 0 at TPSEL[2:0]=010 , PWM2 output is active high. Moreover, the duty cycle and frame rate of PWM2 are both programmable. The duty cycle is determined by registers PWM2DUTY[9:0] (PWM2DUTYH[1:0] and PWM2DUTYL[7:0]). When PWM2DUTY is 0, PWM2 output will be never active. When PWM2DUTY is 0x3FF, PWM2 output will be active for 1023 Timer1 input clocks. The frame rate is determined by Timer1 period. PWM2DUTY value must be less than or equal to Timer1 period. The update of 8F2481 PWM DUTY adopts double buffer mechanism when PWM2 is enabled (PWMxOEN=1) : user write PWM2DUTYL and PWM2DUTYH will not take effect until timer1 overflow. It means new PWM2 Duty cycle can only be available on the next timer1 period.

When PWM2 is disabled, write PWM2DUTY[9:8] MSB 2 bits(PWM2DUTYH[1:0]) first, then write PWM2DUTYL[7:0], this ensure the first timer period gets the right PWM DUTY.

The block diagram of PWM2 is illustrated in the following figure.

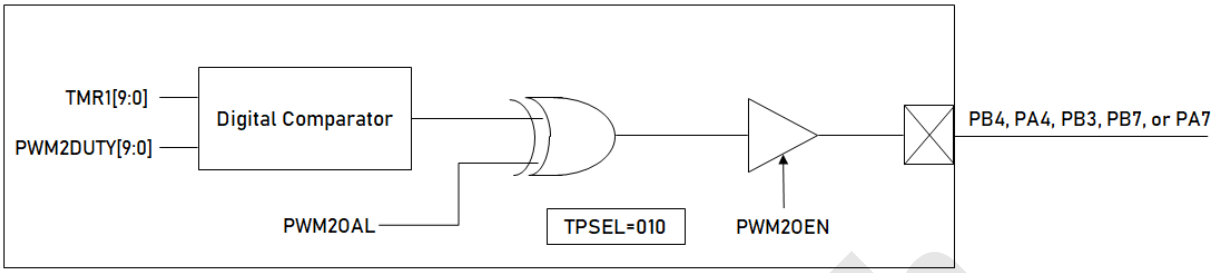


Figure 16 PWM2 Block Diagram

6.3 PWM3

6.3.1 Overview

The PWM3 output can be available on one of the I/O pin PA6, PA2, PB6 or PB0 (selected by configuration word). PWM3 is enabled by setting SELCON(TPSEL[2:0]=011) and PWMxOEN=1 sequentially. PWM3 will become output pin automatically. The active state of PWM3 output is determined by register bit PWMxOAL. When setting PWMxOAL is 1 at TPSEL[2:0]=011, PWM3 output is active low. When PWMxOAL is 0 at TPSEL[2:0]=011, PWM3 output is active high. Moreover, the duty cycle and frame rate of PWM3 are both programmable. The duty cycle is determined by registers PWM3DUTY[9:0] (PWM3DUTYH[1:0] and PWM3DUTYL[7:0]). When PWM3DUTY is 0, PWM3 output will be never active. When PWM3DUTY is 0x3FF, PWM3 output will be active for 1023 Timer1 input clocks. The frame rate is determined by Timer1 period. PWM3DUTY value must be less than or equal to Timer1 period.

The update of 8F2481 PWM DUTY adopts double buffer mechanism when PWM3 is enabled (PWMxOEN=1): user write PWM3DUTYL and PWM3DUTYH will not take effect until timer1 overflow. It means new PWM3 Duty cycle can only be available on the next timer1 period.

When PWM3 is disabled, write PWM3DUTY[9:8] MSB 2 bits(PWM3DUTYH[1:0]) first, then write PWM3DUTYL[7:0], this ensure the first timer period gets the right PWM DUTY.

The block diagram of PWM3 is illustrated in the following figure.

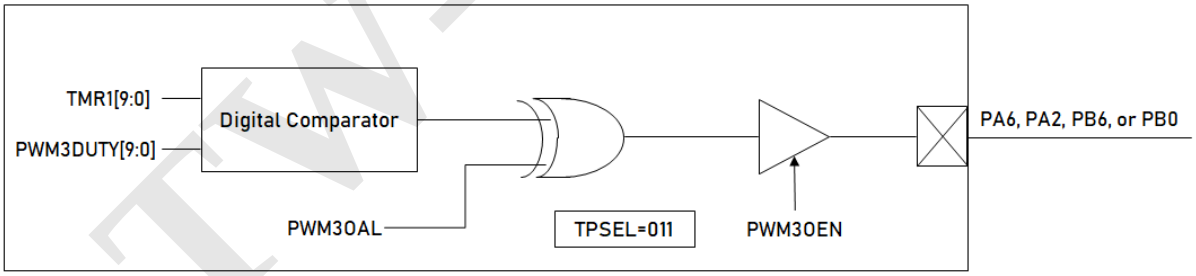


Figure 17 PWM3 Block Diagram

6.4 PWM4

6.4.1 Overview

The PWM4 output can be available on one of the I/O pin PC3, PB3, PA4, PA1 or PC1 (selected by configuration word). PWM4 is enabled by setting SELCON(TPSEL[2:0]=100) and PWMxOEN=1 sequentially. PWM4 will become output pin automatically. The active state of PWM4 output is determined by register bit PWMxOAL. When setting PWMxOAL is 1 at TPSEL[2:0]=100, PWM4 output is active low. When PWMxOAL is 0 at TPSEL[2:0]=100, PWM4 output is active high. Moreover, the duty cycle and frame rate of PWM4 are both programmable. The duty cycle is determined by registers PWM4DUTY[9:0] (PWM4DUTYH[1:0] and PWM4DUTYL[7:0]). When PWM4DUTY is 0, PWM4 output will be never active. When PWM4DUTY is 0x3FF, PWM4 output will be active for 1023 timer4 input clocks. The frame rate is determined by timer4 period. PWM4DUTY value must be less than or equal to timer4 period.

The update of 8F2481 PWM DUTY adopts double buffer mechanism when PWM4 is enabled (PWMxOEN=1): user write PWM4DUTYL and PWM4DUTYH will not take effect until timer4 overflow. It means new PWM4 Duty cycle can only be available on the next timer4 period.

When PWM4 is disabled, write PWM4DUTY[9:8] MSB 2 bits(PWM4DUTYH[1:0]) first, then write PWM4DUTYL[7:0], this ensure the first timer period gets the right PWM DUTY.

The block diagram of PWM4 is illustrated in the following figure.

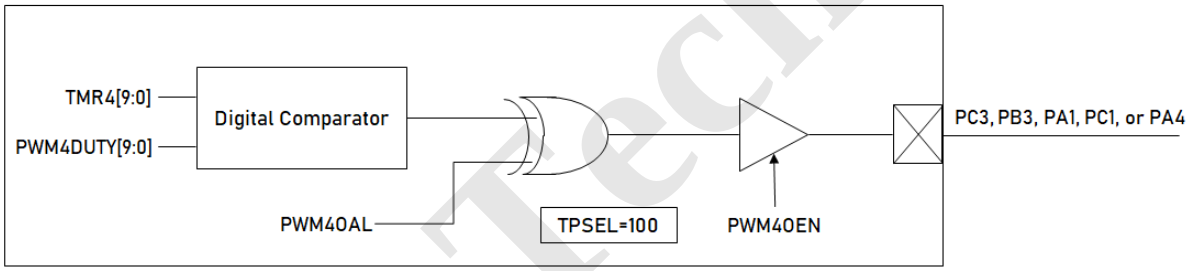


Figure 18 PWM4 Block Diagram

6.5 PWM5

6.5.1 Overview

The PWM5 output can be available on one of the I/O pin PB2, PC0, PC5, PC2 or PA0 (selected by configuration word). PWM5 is enabled by setting SELCON(TPSEL[2:0]=101) and PWMxOEN=1 sequentially. PWM5 will become output pin automatically. The active state of PWM5 output is determined by register bit PWMxOAL. When setting PWMxOAL is 1 at TPSEL[2:0]=101, PWM5 output is active low. When PWMxOAL is 0 at TPSEL[2:0]=101, PWM5 output is active high. Moreover, the duty cycle and frame rate of PWM5 are both programmable. The duty cycle is determined by registers PWM5DUTY[9:0] (PWM5DUTYH[1:0] and PWM5DUTYL[7:0]). When PWM5DUTY is 0, PWM5 output will be never active. When PWM5DUTY is 0x3FF, PWM5 output will be active for 1023 timer5 input clocks. The frame rate is determined by timer5 period. PWM5DUTY value must be less than or equal to timer5 period.

The update of 8F2481 PWM DUTY adopts double buffer mechanism when PWM5 is enabled (PWMxOEN=1): user write PWM5DUTYL and PWM5DUTYH will not take effect until timer5 overflow. It means new PWM5 Duty cycle can only be available on the next timer5 period.

When PWM5 is disabled, write PWM5DUTY[9:8] MSB 2 bits(PWM5DUTYH[1:0]) first, then write PWM5DUTYL[7:0], this ensure the first timer period gets the right PWM DUTY.

The block diagram of PWM5 is illustrated in the following figure.

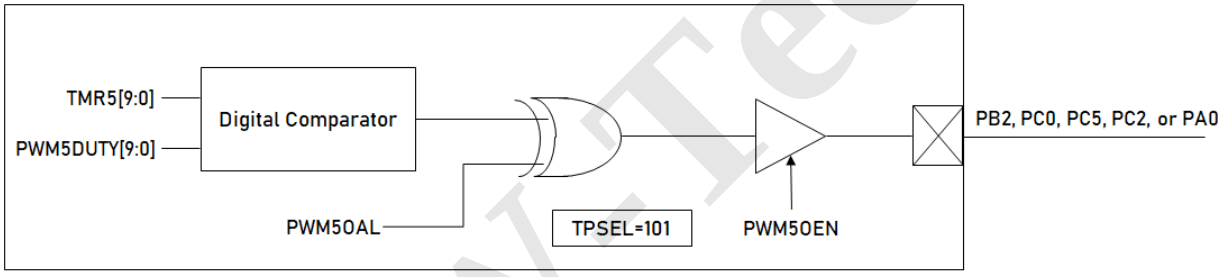


Figure 19 PWM5 Block Diagram

6.6 PWM period

The PWM period is specified by writing to the PRx = [TMRxH[1:0] ,TMRxL[7:0]] register (x=1, 4, 5). The PWM period can be calculated using the following formula:

PWM Period = [(PRx)+1] * $T_{INST\ CYCLE}$ * (TMRx Clock cycle) – PWM clock select F_{INST}

PWM Period = [(PRx)+1] * $T_{HRC\ CYCLE}$ – PWM clock select F_{V_HRC} or F_{HRC}

The PWM frequency is defined as 1/[PWM period]. T_{INST}=1/ F_{INST}, T_{HRC}=1/F_{V_HRC} or 1/ F_{HRC}

6.7 PWM Duty Cycle

The PWM duty cycle (PDx) is specified by writing to the PWMxDUTYH[1:0], PWMxDUTYL[7:0] (PDx, x=1~5) register bits. Up to 10-bit resolution is available. The following equations are used to calculate the PWM duty cycle as a percentage or as time:

PWM Duty Cycle(%) = PDx / (PRx+1) – PWM clock select instruction clock F_{INST}

PWM Duty Cycle(time in s) = PDx * $T_{INST\ CYCLE}$ * (TMRx Clock cycle) – PWM clock select F_{INST}

PWM Duty Cycle(time in s) = PDx * $T_{HRC\ CYCLE}$ – PWM clock select F_{V_HRC} or F_{HRC}

PDx can be written to at any time, but the duty cycle value is not latched into PDx_LH until after TMRx overflow occurs (that is, the period is complete).

The PDx / PWMxDUTYH, PWMxDUTYL registers are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

The maximum PWM resolution (bits) for a given PWM frequency is shown in the following equation:

PWM Resolution (max) = log(PRx+1)/log(2)

Note: If the PWM duty cycle value is longer than the PWM period, the PWM output is not generated.

PWM Frequency	3.906 KHz	7.81 kHz	62.5 kHz	125 kHz	250 KHz	500 KHz
Timer Prescaler	1:2	1:4	1:1	1:1	1:1	1:1
PRx Value	3FFh	FFh	7Fh	3Fh	1Fh	0Fh
Max. Resolution(bits)	10	8	7	6	5	4

Table 6-2 Example PWM Frequencies and Resolutions at 16 MHz/2T Synchronous Mode

6.8 CCP Mode

Key Features of AT8F2481 CCP modules includes

- One 16-bit Input Capture module for a range of edge events on input -- every rising / every falling / 4th rising / 16th rising.
- One 16-bit Output Compare module with multiple output options – Set output / Reset output / trigger interrupt.
- Enhanced PWM: 3 pairs of Half-Bridge with deadband control.
- Enhanced PWM: One Full-Bridge (Forward and Reverse) modes.

6.8.1 Module configuration

When the AT8F2481 CCP modules are enabled, they take over the associated timer and PWM registers, such as PWM5DUTY, PWM4DUTY, TMR5, and TMR4. As a result, the original timer or PWM functions may become unavailable. Care should be taken to avoid enabling other timer-related functions that conflict with CCP operation. The various CCP modes and the registers they occupy are summarized in Table 6-3.

CCP Mode	PWM Registers expropriated	Timer Registers expropriated
Capture	Capture register (16-bit) = {PWM5DUTYL[7:0], PWM4DUTYL[7:0]}	{TMR5L[7:0], TMR4L[7:0]}
Compare	Compare register (16-bit) = {PWM5DUTYL[7:0], PWM4DUTYL[7:0]}	{TMR5L[7:0], TMR4L[7:0]}
PWM(HB/FB)	PWMDUTY register (10-bit) = {PWM5DUTYH[1:0], PWM5DUTYL[7:0]}	{TMR5H[1:0], TMR5L[7:0]}
PWM(HB)	PWMDUTY register (10-bit) = {PWM4DUTYH[1:0], PWM4DUTYL[7:0]}	{TMR4H[1:0], TMR4L[7:0]}

Table 6-3 AT8F2481 CCP Timer and PWM Register Resources

Note: AT8F2481 CCP1 is all-functional, but CCP2, CCP3 has only implement HB functions.

6.8.2 CCP I/O Configuration

CCP modules do not control AT8 pin's direction. The CCP module may have input for capture mode, may have output for compare mode, it also may have up to four PWM outputs (PxA through PxD). The outputs that are active depend on the CCP operating mode selected. The pin assignments are summarized in table 6-4. The proper CCP modes are selected by setting the PM<1:0> and CCPxM<3:0> bits, and pin direction control IOST register also must be initialized.

A suggested start-up sequence for using the CCP module is as follows:

1. Disable the module by writing '0000' to CCPxM<3:0> (CCPxCON<3:0>).

2. Initial pin direction by setting IOST register, initial pin state by IO port register.

3. Setting related timer mode, initial related timer value.

4. Write any initial values to CCP and related register. For Enhanced PWM modes, configure CCPxCON<7:6> as well.

5. Enable the module by writing the appropriate mode select value to CCPxM<3:0>.

6.8.3 CAPTURE MODE (Only Available in CCP1)

In Capture mode, the capture register pair captures the 16-bit value of the selected Timer register when an event occurs on the CCP pin (PB2). An event is defined as one of the following:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

The event is selected by the CCP Mode Select bits, CCPxM<3:0> (CCPxCON<3:0>). When a capture is made, the CCP Interrupt Request Flag bit, CCPIF, is set. (It must be cleared in software.) If another

capture occurs before the value in captured data register is read, the old captured value is overwritten by the new captured value.

Figure 20 shows the Capture mode block diagram.

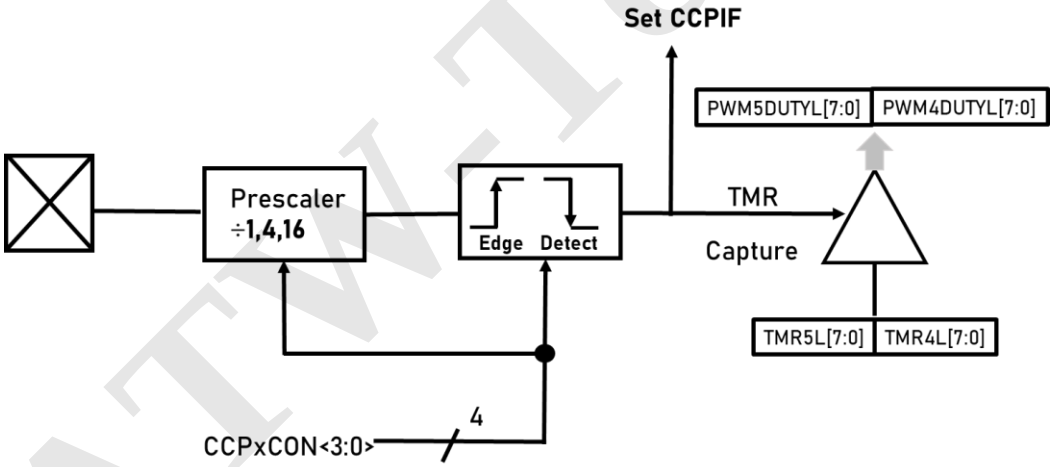


Figure 20 Capture Mode Block Diagram

In Capture mode, the appropriate CCP pin (PB2) should be configured as an input by setting the corresponding IOST direction bit.

To be used for the Capture feature, the selected timers must be initialized and running in Timer mode or Synchronized Counter mode.

Note: The CCP timer is a down count timer. The timer selected to be used with each CCP module as shown in table 6-3.

6.8.4 COMPARE MODE (only Available for CCP1)

In Compare mode, the 16-bit compare data value is constantly compared against the selected timer's register pair value. When a match occurs, the CCP pin can be:

- Driven high
- Driven low
- Toggled (high-to-low or low-to-high)
- Unchanged (that is, reflecting the state of the I/O latch)

The CCP module will not automatically configures the pin as an output when the module is enabled.

To be used for the Compare features, the selected timers must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the Compare operation may not work. Note the CCP timer is running in a down-count mode.

By the way, CCP modules can be equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled the Compare Special Event Trigger mode by selecting CCP1CON (CCPxM<3:0> = 1011).

The Special Event Trigger (CCP_ADC_Start) resets the Timer register pair {TMR5L[7:0], TMR4L[7:0]}

and start an A/D conversion signal (if the A/D module is enabled).

This allows {TMR5L[7:0], TMR4L[7:0]} to serve as a programmable period register.

When the Generate Software Interrupt mode is chosen (CCPxM<3:0> = 1010), the CCP pin is not affected. Only a CCP interrupt is generated, if CCPIE enabled, then the CCPIF bit is set.

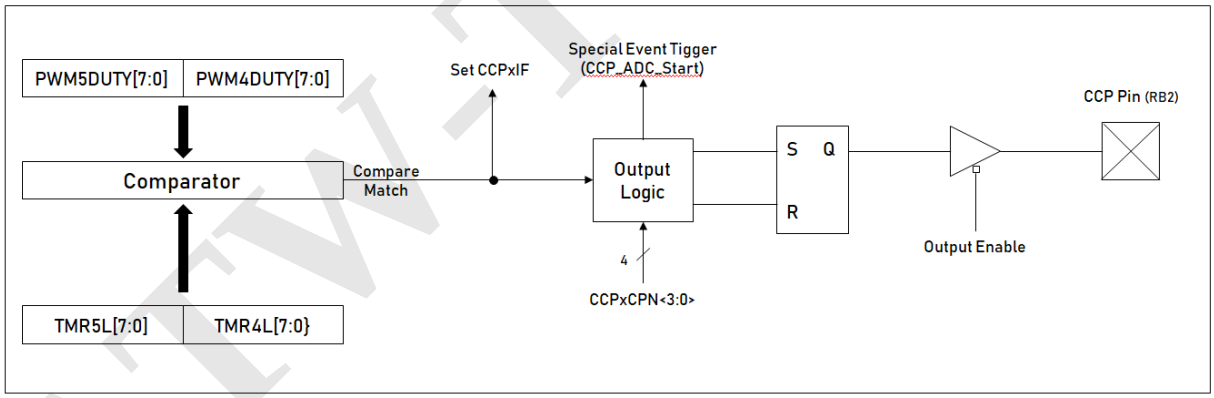


Figure 21 Compare Mode Block Diagram

6.8.5 Enhanced PWM MODE

AT8F2481 CCP module include Enhanced PWM mode in CCP module, which provides another options from the PWM described in section 6. The CCP Enhanced PWM mode is described in section 6.8.6

In CCP Pulse-Width Modulation (PWM) mode, the CCP pin produces a PWM output of up to a 10-bit resolution. Figure 22 shows a simplified block diagram of the module in PWM mode.

A PWM output (Figure 23) has a time-base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

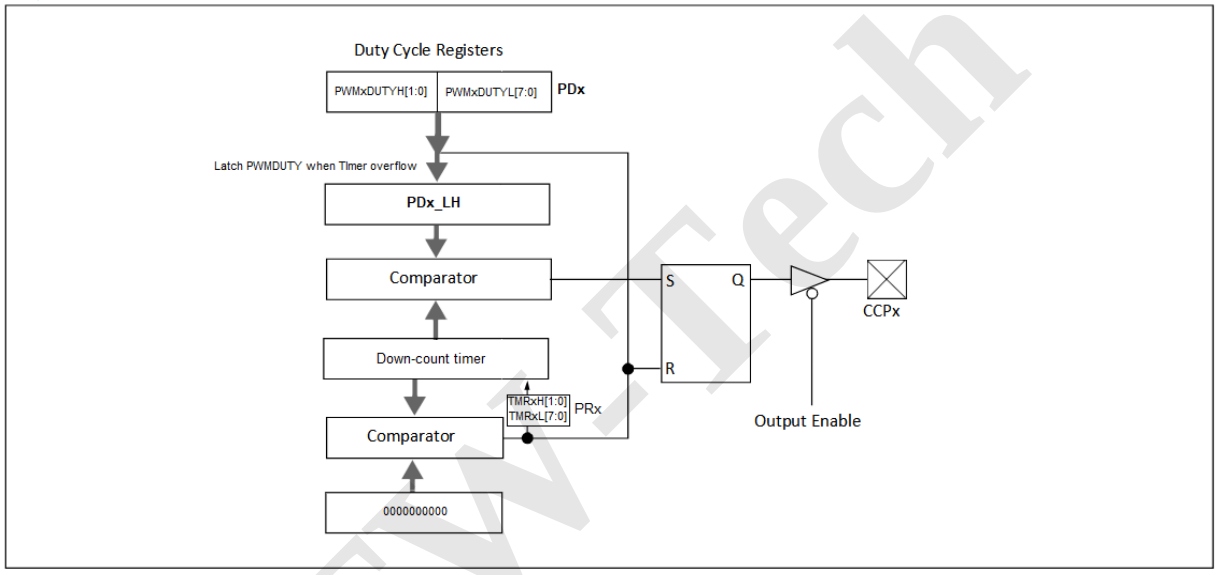


Figure 22 Enhanced PWM Block Diagram

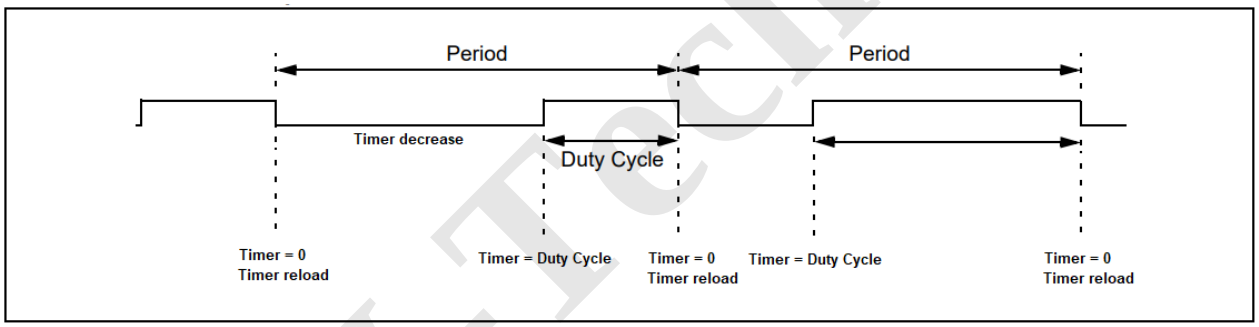


Figure 23 Enhanced PWM Output

6.8.6 Setup for PWM Operation

To configure the CCP module for PWM operation:

- Set proper pin direction and pin initial value.
- Set the PWM period by writing to the PRx register.
- Set the PWM duty cycle by writing to the PDx Register.
- Set the TMRx MODE, prescale value, then enable Timerx by writing to TxCR1.
- Configure the CCPxCON register for PWM operation.

6.8.7 Enhanced PWM Feature

In CCP modules, the Enhanced PWM mode can generate a PWM signal on up to four different output pins, with up to 10 bits of resolution. It can do this through four different PWM Output modes:

1. Single PWM mode (available for CCP1), this function is basic the same as the PWM described in section 6.
2. Half-Bridge PWM mode with deadband control (available for CCP1, CCP2 and CCP3)
3. Full-Bridge PWM, Forward mode (available for CCP1)
4. Full-Bridge PWM, Reverse mode (available for CCP1)

To select an Enhanced PWM mode, the PWMxM<1:0> bits (CCPxCON<7:6>) must be set appropriately.

The PWM outputs are multiplexed with I/O pins, and are designated PxA, PxB, PxC and PxD. Thepolarity of the PWM pins is configurable and is selected by setting the appropriate CCPxM bits in the CCPxCON register provides the pin assignments for each Enhanced PWM mode.

Figure 24 provides an example of a simplified block diagram of the Enhanced PWM module.

Figure 25 provide a first-look of waveforms at various CCP modes.

Note: To prevent the generation of an incomplete waveform when the PWM is first enabled, the CCP module waits until the start of a new PWM period before generating a PWM signal.

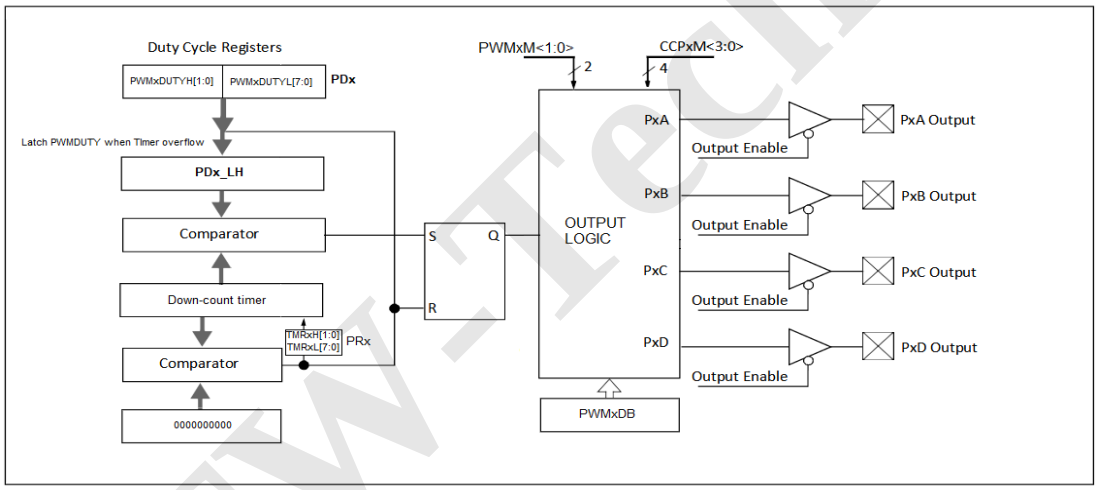


Figure 24 Simplified Block Diagram of Enhanced PWM Mode

ECCP Mode	PWMxM<1:0>	PxA			PxB			PxC			PxD		
CCPx		CCP1	CCP2	CCP3	CCP1	CCP2	CCP3	CCP1	CCP2	CCP3	CCP1	CCP2	CCP3
Single	00	PB2	-	-	-	-	-	-	-	-	-	-	-
Half-Bridge (deadband)	01	PB2/ PC4/ PA7	PC2	PA0	PA5/ PA1/ PA6	PC5	PB3	-	-	-	-	-	-
Full-Bridge	1X	PB2/ PC4/ PA7	-	-	PA5/ PA1/ PA6	-	-	PA2/ PA2/ PA5	-	-	PA3/ PA3/ PA4	-	-

Table 6-4 AT8F2481 CCP Output PAD Configuration

Note:

1. Pad selected by options.

2 . Any pin not used by an Enhanced PWM mode is available for alternate pin functions.

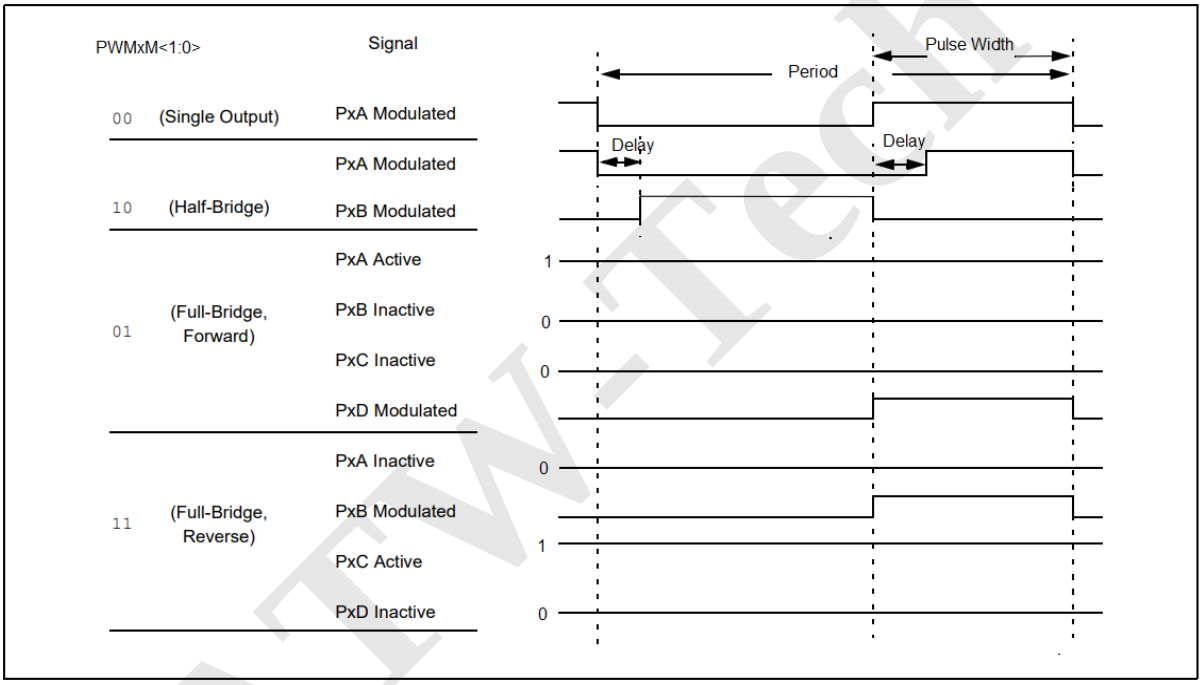


Figure 25 Enhanced PWM Output Relationships (Active-High State) Example

6.8.8 Half-Bridge Mode (available for CCP1, CCP2 and CCP3)

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the PxA pin, while the complementary PWM output signal is output on the PxB pin (see Figure 26). This mode can be used for half-bridge applications, or for full-bridge applications, where four power switches are being modulated with two PWM signals (Figure 27).

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of the DB<7:0> bits of the SFR PWMDBx register sets the number of PWM input clock cycles before the output is driven active.

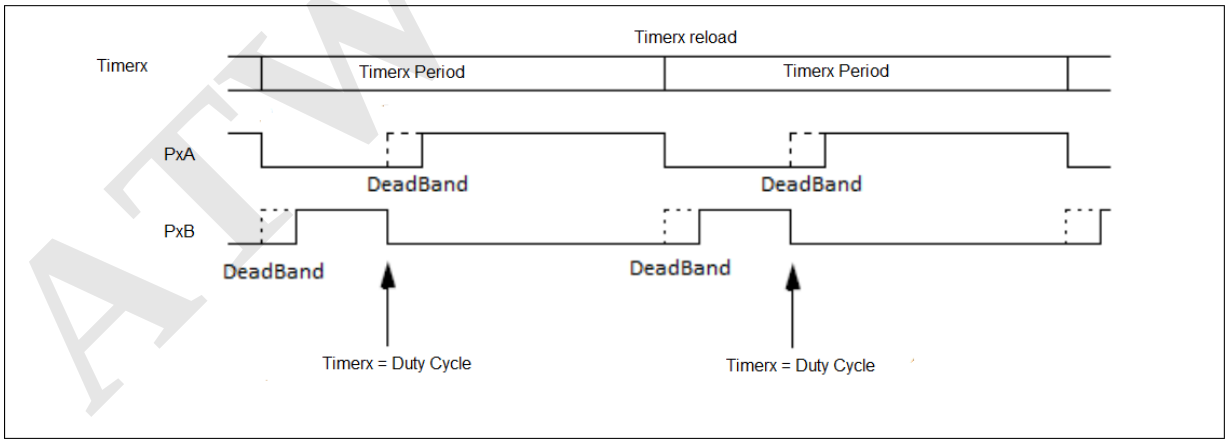


Figure 26 Example of Half-Bridge PWM Output

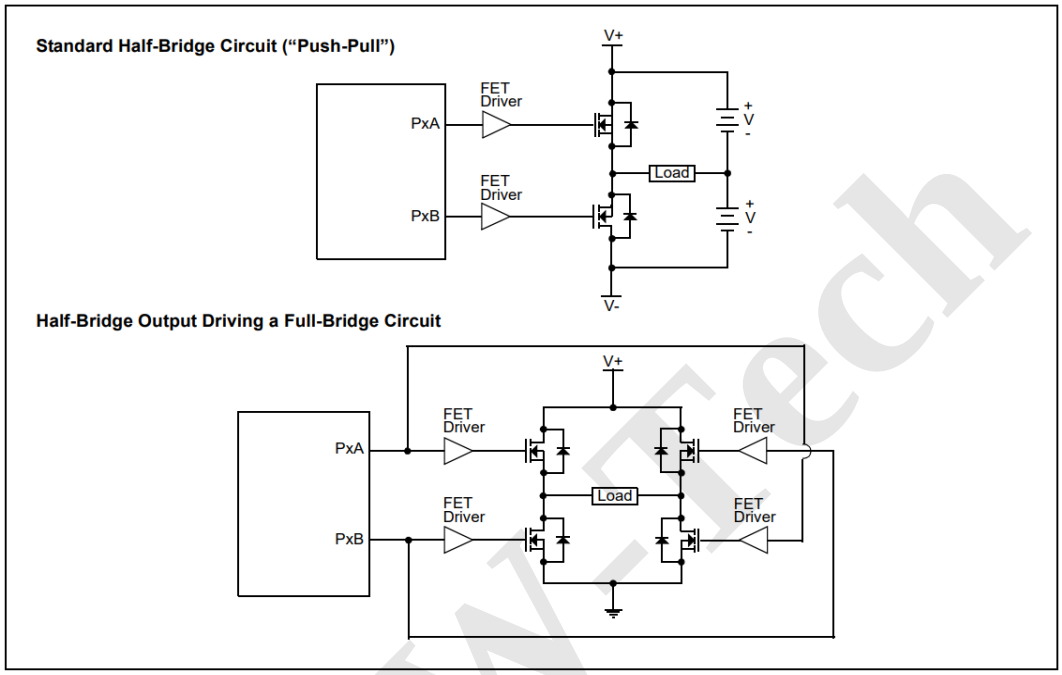


Figure 27 Example of Half-Bridge Applications

6.8.9 Full-Bridge Mode (available for CCP1)

In Full-Bridge mode, all four pins are used as outputs. An example of a full-bridge application is provided in Figure 29.

In the Forward mode, the PxA pin is driven to its active state and the PxD pin is modulated, while the PxB and PxC pins are driven to their inactive state, as shown in Figure 28.

In the Reverse mode, the PxC pin is driven to its active state and the PxB pin is modulated, while the PxA and PxD pins are driven to their inactive state, as shown in Figure 28.

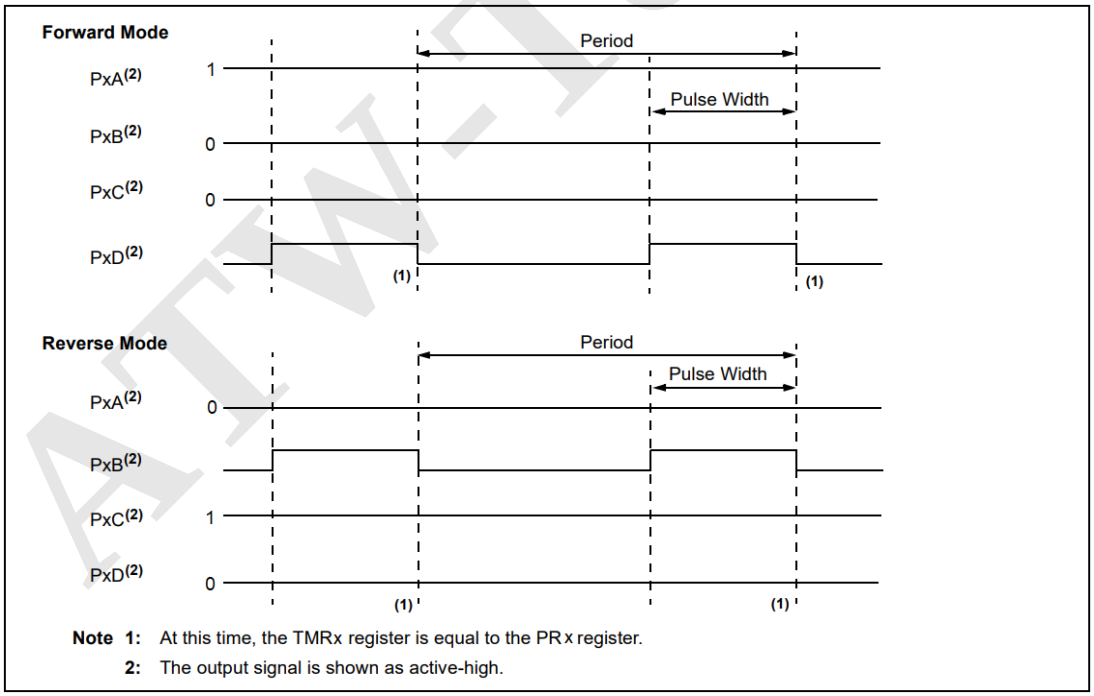


Figure 28 Example of Full-Bridge PWM Output

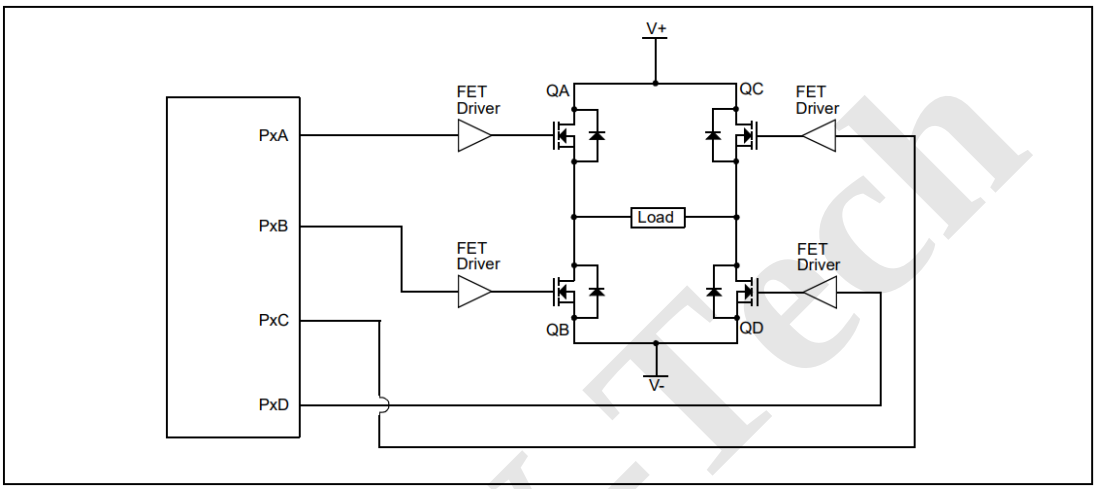


Figure 29 Example of Full-Bridge Applications

6.8.10 Direction Change in Full-Bridge Mode

In the Full-Bridge mode, the PWMxM1 bit in the CCPxCON register allows users to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will change to the new direction on the next PWM cycle.

A direction change is initiated in software by changing the PWMxM1 bit of the CCPxCON register. The following sequence occurs prior to the end of the current PWM period:

Modulation resumes at the beginning of the next period. For an illustration of this sequence, see Figure 30.

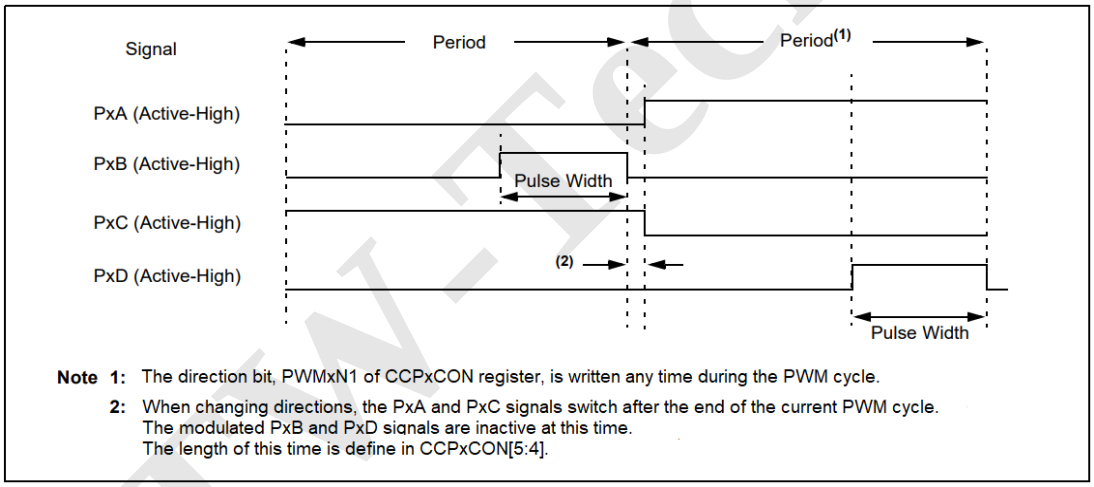


Figure 30 Example of PWM Direction Change

The Full-Bridge mode does not provide a dead-band delay. As one output is modulated at a time, a dead-band delay is generally not required. There is a situation where a dead-band delay is required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near100%.
2. The turn-off time of the power switch, including the power device and driver circuit, isgreater than the turn-on time.

Shows an example of the PWM direction changing from reverse to forward at a near 100% duty cycle. In this example, at time t1, the PxA and PxD outputs become inactive, while thePxC output becomes active. Since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current will flow through power devices, QC and QD, for the duration of 't'. The same phenomenon will occur to power devices, QA and QB, for PWM direction change from reverse to forward.

If an application requires changing PWM direction at high duty cycle, two possible solutions for eliminating the shoot-through current are:

1. Reduce PWM duty cycle for one PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on. Other options to prevent shoot-through current may exist.

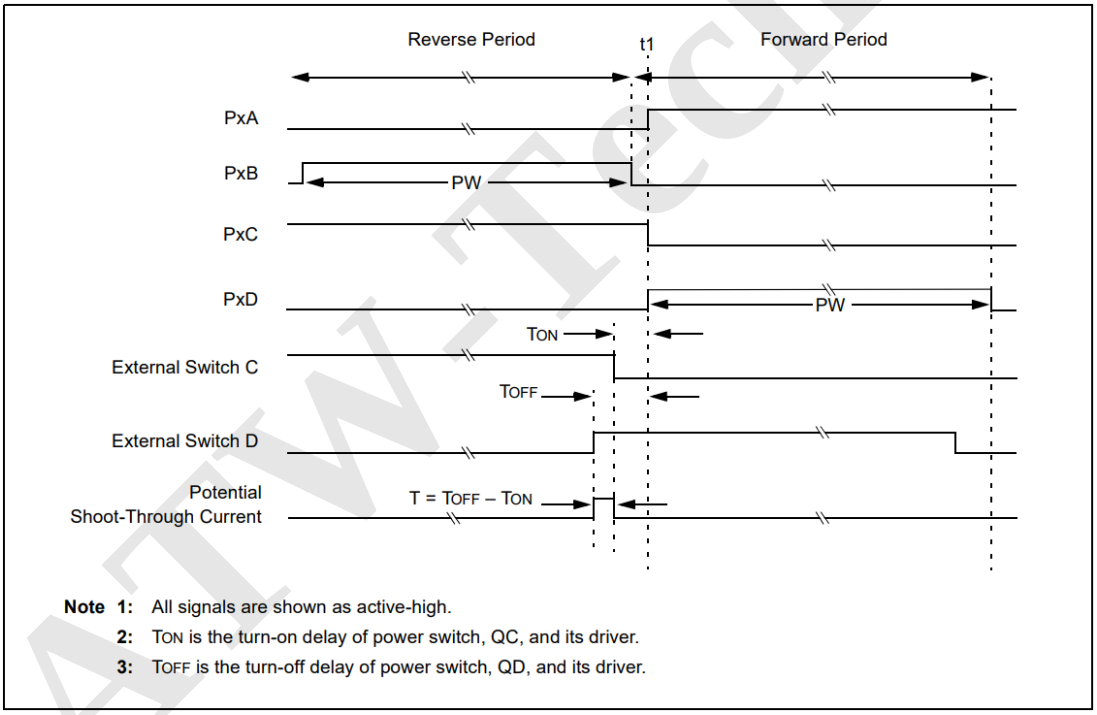


Figure 31 Example of PWM Direction Change at Near 100% Duty Cycle

6.8.11 Start-up Considerations

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

Note: When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the OFF state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

The PWMxM<1:0> bits of the CCPxCON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (PxA/PxC and PxB/PxD). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enabled is not recommended, since it may result in damage to the application circuits.

The PxA, PxB, PxC and PxD output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TxIF bit of the PIR1 register being set, as the second PWM period begins.

6.8.12 Operation in Power Managed Modes

In Sleep mode, all clock sources are disabled. Timers will not decrease and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state.

In Standby mode, the primary clock will continue to clock the module without change.

6.8.13 Effects of A Reset

Any Reset event will force all I/O ports to Input mode and the CCP registers to their Reset states.

6.9 PWM Control Register

6.9.1 PWMxCON (PWM Control Register)

Access the virtual SFR PWMxCON is equivalent to access physical SFR PWM1CON, PWM2CON, PWM3CON, PWM4CON or PWM5CON, according to TPSEL[2:0] value.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxCON	0x1F	-	-	-	-	-	-	PWMxOEN	PWMxOAL
R/W Property		-	-	-	-	-	-	R/W	R/W
Initial Value		x	x	x	x	x	x	0	0

Bit 7:2 **Unimplemented**

For TPSEL [2:0]=001

Bit 1 **PWMxOEN:** Enable/disable PWM1 output

1 = PWM1 output will be present on PAD

0 = PAD is GPIO

Bit 0 **PWMxOAL:** Define PWM1 output active state

1 = PWM1 output is active low

0 = PWM1 output is active high

For TPSEL [2:0]=010

Bit 1 **PWMxOEN:** Enable/disable PWM2 output

1 = PWM2 output will be present on PAD

0 = PAD is GPIO

Bit 0 **PWMxOAL:** Define PWM2 output active state

1 = PWM2 output is active low

0 = PWM2 output is active high

For TPSEL [2:0]=011

Bit 1 **PWMxOEN:** Enable/disable PWM3 output

1 = PWM3 output will be present on PAD

0 = PAD is GPIO

Bit 0 **PWMxOAL:** Define PWM3 output active state

1 = PWM3 output is active low

0 = PWM3 output is active high

For TPSEL [2:0]=100

Bit 1 **PWMxOEN:** Enable/disable PWM4 output

1 = PWM4 output will be present on PAD

0 = PAD is GPIO

Bit 0 **PWMxOAL:** Define PWM4 output active state

1 = PWM4 output is active low

0 = PWM4 output is active high

For TPSEL [2:0]=101

Bit 1 **PWMxOEN:** Enable/disable PWM5 output

1 = PWM5 output will be present on PAD

0 = PAD is GPIO

Bit 0 **PWMxOAL:** Define PWM5 output active state

1 = PWM5 output is active low

0 = PWM5 output is active high

6.9.2 PWMDBx (PWM Dead Band Register)

Access the virtual SFR PWMDBx is equivalent to access physical SFR PWMDB1 or PWMDB2 or PWMDB3, according to SELCON(CCPSEL[1:0]) value.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDBx	0x11E	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

For CCPSEL [1:0]=00

Bit 7:0 **PWM Delay Count for CCP1 Half-Bridge Output Mode:** Number of F_{OSC} or CPU cycles between the P1A transition and the P1B transition.

For CCPSEL [1:0]=10

Bit 7:0 **PWM Delay Count for CCP2 Half-Bridge Output Mode:** Number of F_{OSC} or CPU cycles between the P2A transition and the P2B transition.

For CCPSEL [1:0]=x1

Bit 7:0 **PWM Delay Count for CCP3 Half-Bridge Output Mode:** Number of F_{OSC} or CPU cycles between the P3A transition and the P3B transition.

6.9.3 PWM1DUTY (PWM 1 Duty Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DUTYL	0x91	PWM1DUTY[7:0]							
R/W Property		W							
Initial Value		xxxxxxxx							

Bit 7:0 **PWM1DUTY[7:0]:** PWM1 duty data LSB 8 bits

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DUTYH	0x92	-	-	-	-	-	-	PWM1DUTY[9:8]	
R/W Property		-	-	-	-	-	-	W	
Initial Value		x	x	x	x	x	x	xx	

Bit 7:2 **Unimplemented**

Bit 1:0 **PWM1DUTY[9:8]:** PWM1 duty data MSB 2 bits. (PWM1DUTY[9:0]**)**

6.9.4 PWM2DUTY (PWM 2 Duty Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTYL	0x93	PWM2DUTY[7:0]							
R/W Property		W							
Initial Value		XXXXXXXX							

Bit 7:0 **PWM2DUTY[7:0]:** PWM2 duty data LSB 8 bits

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTYH	0x94	-	-	-	-	-	-	PWM2DUTY[9:8]	
R/W Property		-	-	-	-	-	-	W	
Initial Value		x	x	x	x	x	x	xx	

Bit 7:2 **Unimplemented**

Bit 1:0 **PWM2DUTY[9:8]:** PWM2 duty data MSB 2 bits. (PWM2DUTY[9:0]**)**

6.9.5 PWM3DUTY (PWM 3 Duty Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTYL	0x95	PWM3DUTY[7:0]							
R/W Property		W							
Initial Value		xxxxxxxx							

Bit 7:0 **PWM3DUTY[7:0]:** PWM3 duty data LSB 8 bits

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTYH	0x96	-	-	-	-	-	-	PWM3DUTY[9:8]	
R/W Property		-	-	-	-	-	-	W	
Initial Value		x	x	x	x	x	x	xx	

Bit 7:2 **Unimplemented**

Bit 1:0 **PWM3DUTY[9:8]:** PWM3 duty data MSB 2 bits. (PWM3DUTY[9:0]**)**

6.9.6 PWM4DUTY (PWM 4 Duty Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTYL	0x97	PWM4DUTY[7:0]							
R/W Property		W							
Initial Value		xxxxxxxx							

Bit 7:0 **PWM4DUTY[7:0]:** PWM4 duty data LSB 8 bits

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTYH	0x98	-	-	-	-	-	-	PWM4DUTY[9:8]	
R/W Property		-	-	-	-	-	-	W	
Initial Value		x	x	x	x	x	x	xx	

Bit 7:2 **Unimplemented**

Bit 1:0 **PWM4DUTY[9:8]:** PWM4 duty data MSB 2 bits. (PWM4DUTY[9:0]**)**

6.9.7 PWM5DUTY (PWM 5 Duty Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5DUTYL	0x99	PWM5DUTY[7:0]							
R/W Property		W							
Initial Value		xxxxxxxx							

Bit 7:0 **PWM5DUTY[7:0]:** PWM5 duty data LSB 8 bits

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5DUTYH	0x9A	-	-	-	-	-	-	PWM5DUTY[9:8]	
R/W Property		-	-	-	-	-	-	W	
Initial Value		x	x	x	x	x	x	xx	

Bit 7:2 **Unimplemented**

Bit 1:0 **PWM5DUTY[9:8]:** PWM5 duty data MSB 2 bits. (PWM5DUTY[9:0])

6.9.8 CCPxCON (CCPx Control Register)

Access the virtual SFR CCPxCON is equivalent to access physical SFR CCP1CON or CCP2CON or CCP3CON, according to SELCON(CCPSEL[1:0]) value.

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPxCON	0x11F	PWMx M1	PWMx M0	FBCH1	FBCH0	CCPx M3	CCPx M2	CCPx M1	CCPx M0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

When CCPSEL[1:0] = 00, access physical SFR CCP1CON

Bit 7:6 **PWMxM[1:0]:** (available for CCPxM[3:2]=11**)**

00 = PWM Single output.

01 = PWM Full-bridge output forward

10 = PWM Half-bridge output

11 = PWM Full-bridge output reverse

Bit 5:4 **FBCH[1:0] :** Full band change direction gap

00 = 1 CPU Clock

01 = 4 CPU Clock

1x = 16 CPU Clock

Bit 3:0 **CCPxM[3:0]:** CCP mode select

0000 = OFF

0010 = Compare mode, toggle output on match.

0100 = Capture mode, capture at every falling edge.

0101 = Capture mode, capture at every rising edge.

0110 = Capture mode, capture at every 4th rising edge.

0111 = Capture mode, capture at every 16th rising edge.

1000 = Compare mode, set output on match.

1001 = Compare mode, clear output on match.

1010 = Compare mode, interrupt on match.

1011 = Compare mode, trigger special event.

1100 = PWM mode, P1A/P1C active high, P1D/P1B active high

1101 = PWM mode, P1A/P1C active high, P1D/P1B active low

1110 = PWM mode, P1A/P1C active low, P1D/P1B active high
1111 = PWM mode, P1A/P1C active low, P1D/P1B active low

When CCPSEL[1:0] = 10, access physical SFR CCP2CON

Bit 7:6 **PWMxM[1:0]:** (available for CCPxM[3:2]=11**)**
00 = N.C.
01 = N.C.
10 = N.C.
11 = PWM Half-bridge output

Bit 3:0 **CCPxM[3:0]:** CCP mode select
1100 = PWM mode, P2A active high, P2B active high
1101 = PWM mode, P2A active high, P2B active low
1110 = PWM mode, P2A active low, P2B active high
1111 = PWM mode, P2A active low, P2B active low

When CCPSEL[1:0] = x1, access physical SFR CCP3CON

Bit 7:6 **PWMxM[1:0]:** (available for CCPxM[3:2]=11**)**
00 = N.C.
01 = N.C.
10 = N.C.
11 = PWM Half-bridge output

Bit 3:0 **CCPxM[3:0]:** CCP mode select
1100 = PWM mode, P3A active high, P3B active high
1101 = PWM mode, P3A active high, P3B active low
1110 = PWM mode, P3A active low, P3B active high
1111 = PWM mode, P3A active low, P3B active low

7. Buzzer

7.1 Overview

The Buzzer1 output (BZ1) can be available on I/O pin PB3 when register bit BZ1EN(BZ1CR[7]) is set to 1. Moreover, PB3 will become output pin automatically. The frequency of BZ1 can be derived from Timer1 output or Prescaler1 output and dividing rate is determined by register bits BZ1FSEL[3:0](BZ1CR[3:0]). When BZ1FSEL[3] is 0, Prescaler1 output is selected to generate BZ1 output. When BZ1FSEL[3] is 1, Timer1 output is selected to generate BZ1 output. The dividing rate can be from 1:2 to 1:256 in order to generate all kinds of frequency. The block diagram of Buzzer1 is illustrated in the following figure

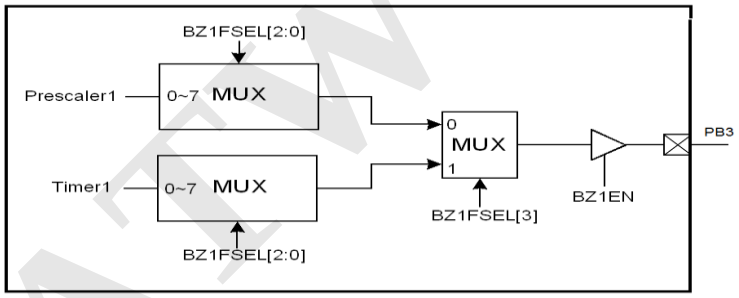


Figure 32 Buzzer1 Block Diagram

7.2 BZ1CR (Buzzer1 Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ1CR	0x13	BZ1EN	-	-	-	BZ1FSEL[3:0]			
R/W Property		W	-	-	-	W			
Initial Value		0	x	x	x	1	1	1	1

Bit 7 **BZ1EN:** Enable/Disable BZ1 output

1 = Enable Buzzer1

0 = Disable Buzzer1

Bit 6:4 **Unimplemented**

Bit 3:0 **BZ1FSEL[3:0]:** Frequency selection of BZ1 output

BZ1FSEL[3:0]	BZ1 Frequency Selection	
	Clock Source	Dividing Rate
0000	Prescaler1 output	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000	Timer1 output	Timer1 bit 0
1001		Timer1 bit 1
1010		Timer1 bit 2
1011		Timer1 bit 3
1100		Timer1 bit 4
1101		Timer1 bit 5
1110		Timer1 bit 6
1111		Timer1 bit 7

8. IR Carrier

8.1 Overview

The IR carrier pad can be PB1 or PA3 selected by configuration word. The IR carrier will be generated after register bit IREN (IRCR[0]) is set to 1. When IREN is set to 1, the IR carrier pad will become output pin automatically. When IREN is clear to 0, PB1 or PA3 will become general I/O pin as it was configured.

The IR carrier frequency is selectable by register bit IRF57K (IRCR[1]). When IRF57K is 1, IR carrier frequency is 57KHz. When IRF57K is 0, IR carrier frequency is 38KHz. Because IR carrier frequency is derived from high frequency system oscillation F_{HOSC}, it is necessary to specify what frequency is used as system oscillation when external crystal is used. Register bit IROSC358M (IRCR[7]) is used to provide AT8F2481 this information. When IROSC358M is 1, frequency of external crystal is 3.58MHz and when IROSC358M is 0, frequency of external crystal is 455KHz. When internal high frequency oscillation is adopted, this register will be ignored, and it will provide 4MHz clock to IR module.

The active state (polarity) of IR carrier is selectable according to IR carrier pad output data. When register bit IRCSEL (IRCR[2]) is 1, IR carrier will be present on IR carrier pad when its output data is 0. When register bit IRCSEL (IRCR[2]) is 0, IR carrier will be present on IR carrier pad when its output data is 1. The polarity of IR carrier is shown in the following figure.

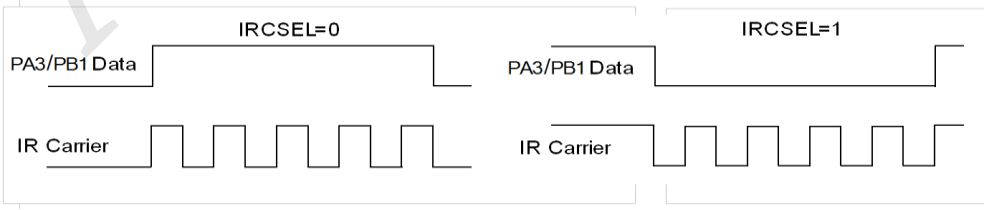


Figure 33 Polarity of IR Carrier vs. Output Data

IREN Register	IR_PAD Option	IR function	IR Pad
0	X	OFF	-
1	1	ON	PA3
1	0	ON	PB1

Table 8-1 IR Carrier Selection

8.2 IRCR (IR Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCR	0x90	IROSC358M	-	-	-	-	IRCSEL	IRF57K	IREN
R/W Property		W	-	-	-	-	W	W	W
Initial Value		0	x	x	x	x	0	0	0

Bit 7 **IROSC358M:** When external crystal is used, this bit is determined according to what kind of crystal is used. This bit is ignored if internal high frequency oscillation is used.
1 = Crystal frequency is 3.58MHz
0 = Crystal frequency is 455KHz

Bit 6:3 **Unimplemented**

Bit 2 **IRCSEL:** Polarity selection of IR carrier
1 = IR carrier will be generated when I/O pin data is 0
0 = IR carrier will be generated when I/O pin data is 1

Bit 1 **IRF57K:** Polarity selection of IR carrier
1 = IR carrier frequency is 57KHz
0 = IR carrier frequency is 38KHz

Bit 0 **IREN:** Polarity selection of IR carrier
1 = Enable IR carrier output
0 = Disable IR carrier output

Note:

1. Only high oscillation (F_{HOSC}) (See section 15. Oscillation Configuration) can be used as IR clock source.

2. Division ratio for different oscillation type.

OSC. Type	57KHz	38KHz	Conditions
High IRC(4MHz)	64	96	HIRC mode (the input to IR module is set to 4MHz no matter what I_HRC system clock is)
Xtal 3.58MHz	64	96	Xtal mode & IROSC358M=1
Xtal 455KHz	8	12	Xtal mode & IROSC358M=0

Table 8-2 Division ratio for different oscillation type

9. RFC

9.1 Overview

AT8F2481 has built-in RFC mode. Once RFC mode is enabled, the selected input pad state will take control *of the Timer1 counting. When the selected input pad is recognized as 0 state (The input pad voltage is* smaller than V_{IL}), Timer1 keeps counting. When this selected pad is recognized as 1 (The input pad voltage is larger than V_{IH}), Timer1 stops counting. The following figure shows how RFC mode operates: PSEL3~0 is used to select one RFC input pad out of 16 AT8F2481 pads. RFCEN is used to switch the Timer1 enable signal between the normal enable signal T1EN and RFC selected input state.

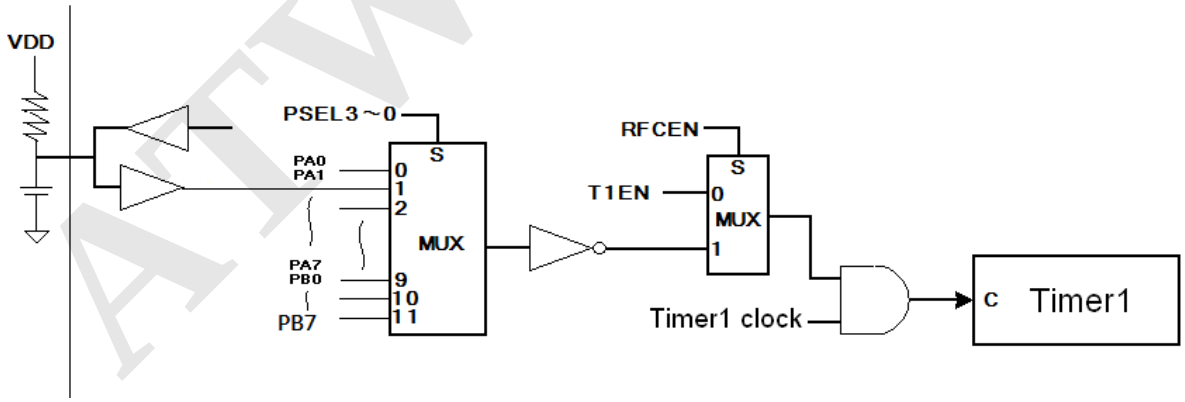


Figure 34 RFC Block Diagram

One application of RFC mode is to measure the capacitor-resistor charging time, As the figure shows, when PSEL3~0=0x01, PA1 is selected as RFC input pad. At first the PA1 is set as output low (the voltage of PA1 is discharged to 0). Next step, clear Timer1 content, set PA1 as input and enable RFC mode. Then Timer1 will start counting, and the RC circuit will start charging PA1. As PA1 is charged to the VIH voltage, the Timer1 counting is stopped because PA1 input is high. The Timer1 content will show the RC circuit charging time. (Note: Timer1 is down-count.)

9.2 RFC (RFC Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFC	0x189	RFCEN	-	-	-	PSEL[3:0]			
R/W Property		R/W	-	-	-	R/W			
Initial Value		0	x	x	x	0			

Bit 7 **RFCEN:** Enable/disable RFC function.
1 = Enable RFC function
0 = Disable RFC function

Bit 6:4 **Unimplemented**

Bit 3:0 **PSEL[3:0]:** Select RFC PAD

PSEL[3:0]	RFC PAD
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100	PA4
0101	PA5
0110	PA6
0111	PA7
1000	PB0
1001	PB1
1010	PB2
1011	PB3
1100	PB4
1101	PB5
1110	PB6
1111	PB7

10.Low Voltage Detector (LVD)

10.1 Overview

The AT8F2481 low voltage detector (LVD) built-in precise band-gap reference for accurately detecting VDD level. If LVDEN (register PCON[5])=1 and CMP_INV (register CMPCR[4])=0) and VDD voltage value falls below LVD voltage which is selected by RBIAS_H,RBIAS_L , LVDS[3:0] as table shown below, the LVD output will become low. If the LVD interrupt is enabled, the LVD interrupt flag will be high and if GIE=1 it will force the program to execute interrupt service routine. Moreover, LVD real-state output can be polled by register LVDCON[6]. IF CMP_INV (register CMPCR[4])=1) , the LVD interrupt flag will be high at the condition of VDD voltage value rising beyond LVD voltage.

The following is LVD block diagram:

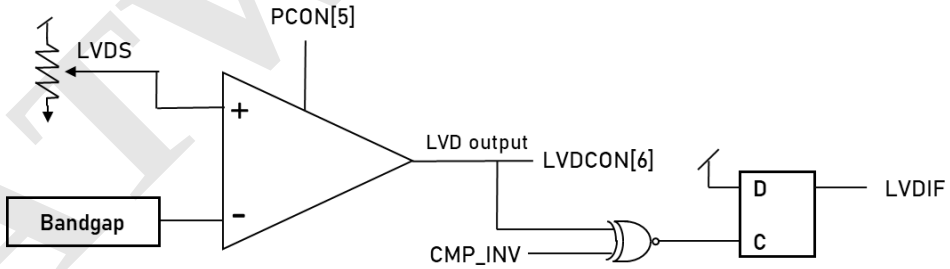


Figure 35 LVD Block Diagram

10.2 LVD Control Register

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCON	0x89	-	LVDOUT	-	-	LVDS3	LVDS2	LVDS1	LVDS0
R/W Property		-	R	-	-	R/W	R/W	R/W	R/W
Initial Value		x	x	x	x	1	1	1	1

Bit 7 **Unimplemented**

Bit 6 **LVDOUT:** Low voltage detector output, read-only.

Bit 5:4 **Unimplemented**

Bit 3:0 **LVDS[3:0]:** Select LVD voltage

The voltage threshold can be adjusted by configuring RBIAS_H, RBIAS_L, and LVDS[3:0]. The available monitoring levels are detailed in the table as below. RBIAS_H and RBIAS_L refers to 11.2.2.

Note:

1. The hysteresis voltage VDD ↑ (from low to high) of LVD is about 0.1V.

2. In battery charging applications (detected voltage is from low to high), the LVD voltage select table should be as followed:

RBIAS_H	RBIAS_L	LVDS[3:0]	VDD ↓	VDD ↑
0	1	0110	4.84V	(4.84+0.1) V
0	1	0101	4.54V	(4.54+0.1) V
1	1	1011	4.39V	(4.39+0.1) V
0	0	1111	4.15V	(4.15+0.1) V
0	0	1110	4.05V	(4.05+0.1) V

0	0	1101	3.90V	(3.90+0.1) V
0	0	1100	3.75V	(3.75+0.1) V
0	0	1011	3.60V	(3.60+0.1) V
0	1	0011	3.50V	(3.50+0.1) V
0	0	1010	3.45V	(3.45+0.1) V
0	0	1001	3.30V	(3.30+0.1) V
0	0	1000	3.15V	(3.15+0.1) V
0	1	0010	3.05V	(3.05+0.1) V
0	0	0111	3.00V	(3.00+0.1) V
0	0	0110	2.90V	(2.90+0.1) V
0	0	0101	2.80V	(2.80+0.1) V
0	1	0001	2.64V	(2.64+0.1) V
0	0	0100	2.60V	(2.60+0.1) V
1	0	1110	2.52V	(2.52+0.1) V
0	1	0000	2.46V	(2.46+0.1) V
0	0	0011	2.40V	(2.40+0.1) V
1	0	1100	2.33V	(2.33+0.1) V
1	0	1011	2.24V	(2.24+0.1) V
0	0	0010	2.20V	(2.20+0.1) V
1	0	1001	2.05V	(2.05+0.1) V
0	0	0001	2.00V	(2.00+0.1)V
0	0	0000	1.90V	(1.90+0.1)V

The LVD control flow is as the following:

Step1: Select LVD voltage by LVDS[3:0]

Step2: Set CMPCR (CMP_INV=0; CMPCR[3:0]=0xA)

Step3: PCON[5]=1 (enable LVD)

Step4: Check LVD status by LVDCON[6]

Note: If LVD voltage LVDS[3:0] is changed, user must wait at least 50us(@F_{HOSC}=1MHz) to get correct LVD status by LVDCON[6]

Example:

```
    MOVIA       LVD_4P15V
    MOVAR       LVDCON                          ; Select LVD voltage
    MOVIA       RBias_High_Dis | RBias_Low_Dis | CMPFINV_Dis | 0x0A
    MOVAR       CMPCR                           ; Set CMPCR (CMP_INV=0; CMPCR[3:0]=0xA)
    BSR         PCON,LVDEN                      ; Enable LVD
L_MAIN_LOOP:
    CLRWDT                                      ; Clear watch dog
    BTRSC       LVDCON,LVDOUT                   ; If VDD < LVD, PB0=0, otherwise PB0=1
    BSR         PORTB,0
    BTRSS       LVDCON,LVDOUT
    BCR         PORTB,0
    LGOTO       L_MAIN_LOOP
```

11.Voltage Comparator (CMP)

11.1 Overview

AT8F2481 provides 1 set of voltage comparator and internal reference voltage with various analog comparing mode. The comparator non-inverting and inverting input can share with GPIO. The internal reference voltage can only routed to inverting input of comparator.

CMPEN (register CMPCON[7]) is used to enable and disable comparator. When CMPEN=0(default), comparator is disabled. When CMPEN=1, the comparator is enabled. In halt mode the comparator is disabled automatically. The structure of comparator is shown in the following figure:

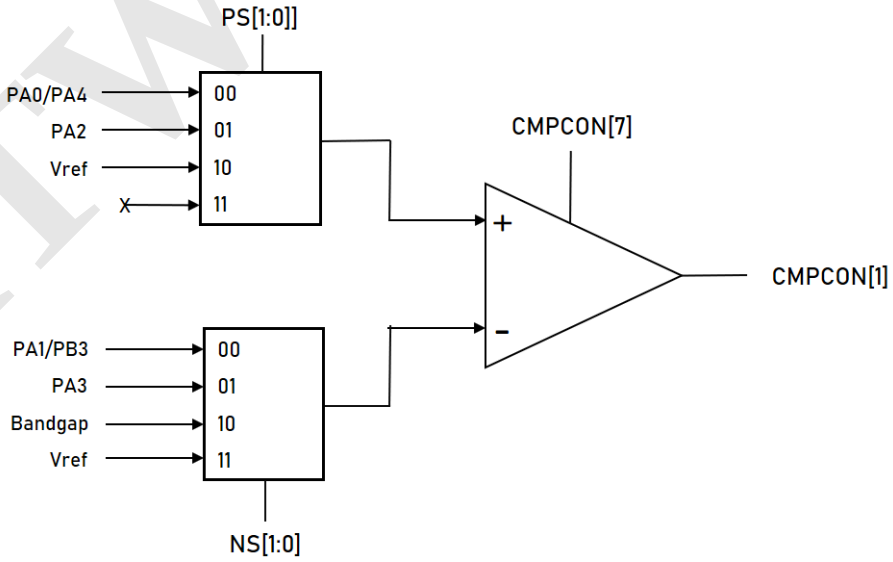


Figure 36 Comparator Hardware Connection

Note: P2P mode can option (PA.0/PA.1/PA.2/PA.3) or (PA.2/PA.3/PA.4/PB.3).

11.2 Comparator Reference Voltage (Vref)

The internal reference voltage Vref is built by series resistance to provide different level of reference voltage. RBIAS_H and RBIAS_L are used to select the maximum and minimum values of Vref, and LVDS[3:0] are used to select one of 16 voltage levels.

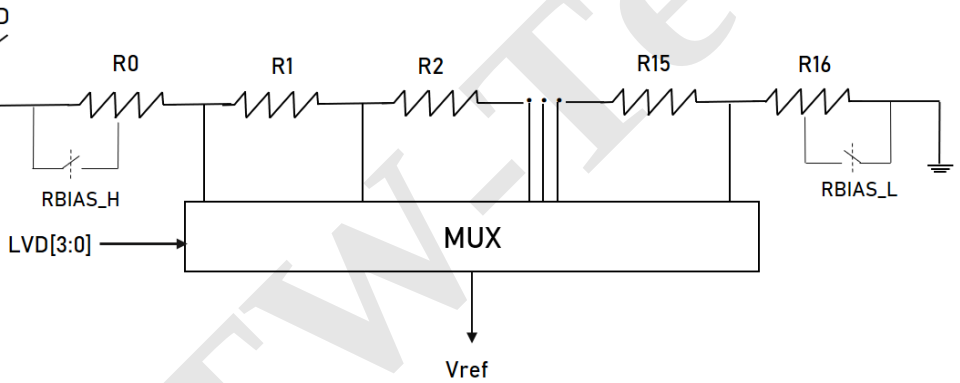


Figure 37 Vref Hardware Connection

The Vref is determined by RBIAS_H, RBIAS_L and LVDS[3:0]. The LVDS[3:0] is used to select one out of 16 reference voltages, the table shown below.

RBIAS_H	RBIAS_L	LVDS[3:0]	V_{IN}-=Vref	Unit
1	0	0000	0.508*VDD	V
1	0	0001	0.483*VDD	
1	0	0010	0.439*VDD	
1	1	0000	0.420*VDD	
1	0	0011	0.402*VDD	
1	1	0001	0.390*VDD	
1	0	0100	0.371*VDD	
1	0	0101	0.345*VDD	
1	0	0110	0.333*VDD	
1	0	0111	0.322*VDD	
1	0	1000	0.306*VDD	
0	0	0001	0.300*VDD	
1	0	1001	0.292*VDD	
1	0	1010	0.280*VDD	
0	0	0010	0.273*VDD	
1	0	1011	0.268*VDD	
1	0	1100	0.257*VDD	
1	0	1101	0.247*VDD	
1	0	1110	0.238*VDD	
1	0	1111	0.233*VDD	
0	1	0001	0.227*VDD	
1	1	0110	0.213*VDD	
0	0	0110	0.207*VDD	
1	1	0111	0.200*VDD	
0	0	1000	0.190*VDD	
0	0	1001	0.182*VDD	
0	0	1010	0.174*VDD	
0	0	1011	0.167*VDD	
0	0	1100	0.160*VDD	
0	0	1101	0.154*VDD	
0	0	1110	0.148*VDD	
1	1	1011	0.137*VDD	
0	1	0101	0.132*VDD	
0	1	0110	0.124*VDD	
0	1	0111	0.116*VDD	
0	1	1000	0.106*VDD	
1	1	1110	0.102*VDD	
0	1	1001	0.096*VDD	
0	1	1010	0.088*VDD	
0	1	1011	0.080*VDD	
0	1	1100	0.072*VDD	
0	1	1101	0.065*VDD	
0	1	1110	0.059*VDD	
0	1	1111	0.055*VDD	

Table 11-1 The Reference Voltage Vref Selection Table

There are two ways to get the comparator output result: one is through register polling, the other is through probing output pad.

Comparator output can be polled by CMPOUT (register CMPCON[1]).

To probe comparator output at output pad, set CMPOE (register CMPCON[0]) to 1, then PB3 will be the real-time state of the comparator output. It is noted that when CMPOE=1, the PWM3 function will be disabled if it is enabled.

11.3 Comparator Control Register

11.3.1 CMPCON (Comparator Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCON	0x9C	CMPEN	BIASEN	-	-	-		CMPOUT	CMPOE
R/W Property		R/W	R/W	-	-	-		R	R/W
Initial Value		0	0	x	x	xx		x	0

Bit 7 **CMPEN:** Enable/Disable comparator
1 = Enable comparator
0 = Disable comparator

Bit 6 **BIASEN:** Comparator / LVD module bias enable
1 = Comparator/LVD bias is enable
0 = Comparator/LVD bias is disable

Bit 5:2 **Unimplemented**

Bit 1 **CMPOUT:** Comparator output status, read-only.

Bit 0 **CMPOE:** Disable/enable comparator output to pad PB3
1 = Enable comparator output to pad PB3
0 = Disable comparator output to pad PB3

Note: Comparator output to pad PB3 has higher priority than buzzer1 output.

11.3.2 CMPCR (Comparator Voltage Select Control Register)

Name	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	0x9D	-	RBIAS_H	RBIAS_L	CMP_INV	PS1	PS0	NS1	NS0
R/W Property		-	R/W						
Initial Value		x	0	0	0	1	1	0	0

Bit 7 **Unimplemented**

Bit 6:5 **RBIAS[H:L]:** Set corresponding voltage reference levels

Bit 4 **CMPF_INV:** Comparator output inverse control bit

Bit 3:2 **PS[1:0]:** Comparator non-inverting input select

PS[1:0]	Non-inverting input
00	PA0
01	PA2
10	Vref
11	---

Bit 1:0 **NS[1:0]:** Comparator inverting input select

NS[1:0]	Non-inverting input
00	PA1
01	PA3
10	Bandgap (0.6V)
11	Vref

Example: (For P2P Mode)

```
    MOVIA       PA0_Input | PA1_Input
    MOVAR       IOSTA                           ; Set PA0 & PA1 as input pin
    MOVIA       RBias_High_Dis | RBias_Low_Dis| CMPFINV_Dis
    MOVAR       CMPCR                           ; Set PA0 is non-invert input, PA1 is invert input
    BSR         CMPCON,CMPEN                    ; Enable comparator
    BCR         PIR2, CMPIF                     ; Clear comparator interrupt flag
    BSR         PIE2, CMPIE                     ; Enable comparator interrupt
    BSR         INTCON, PEIE                    ; Enable peripheral interrupt
    BSR         INTCON,GIE                      ; Enable Global Interrupt
L_MAIN_LOOP:
    CLRWDT                                      ; Clear watch dog
    BTRSC       CMPCON,CMPOUT                   ; If PA0 > PA1, PB7=1 (CMPOUT=1)
    BSR         PORTB,7
    BTRSS       CMPCON,CMPOUT                   ; If PA0 < PA1, PB7=0 (CMPOUT=0)
    BCR         PORTB,7
    LGOTO       L_MAIN_LOOP
ISR:
    BTRSS       PIR2, CMPIF                     ; Check comparator interrupt flag
    LGOTO       ISR_Exit
    MOVIA       0x40
    XORAR       PORTB,1                         ; PB6 toggle output, when compare occur
    BCR         PIR2, CMPIF                     ; Clear comparator interrupt flag
ISR_Exit:
    RETIE
```

Example: (For P2V Mode)

```
    MOVIA       PA0_Input
    MOVAR       IOSTA                           ; Set PA0 & PA1 as input pin
    MOVIA       RBias_High_En | RBias_Low_Dis| CMPFINV_Dis
    MOVAR       CMPCR                           ; Set PA0 is non-invert input, VREF is invert input
    BSR         CMPCON,CMPOE                    ; Enable CMPOUT status output to PB3
    BSR         CMPCON,CMPEN                    ; Enable comparator
    BCR         PIR2, CMPIF                     ; Clear comparator interrupt flag
    BSR         PIE2, CMPIE                     ; Enable comparator interrupt
    BSR         INTCON, PEIE                    ; Enable peripheral interrupt
    BSR         INTCON,GIE                      ; Enable Global Interrupt
L_MAIN_LOOP:
    CLRWDT                                      ; Clear watch dog
    BTRSC       CMPCON,CMPOUT                   ; If PA0 > VREF, PB7=1 (CMPOUT=1)
    BSR         PORTB,7
    BTRSS       CMPCON,CMPOUT                   ; If PA0 < VREF, PB7=0 (CMPOUT=0)
    BCR         PORTB,7
    LGOTO       L_MAIN_LOOP
ISR:
    BTRSS       PIR2, CMPIF                     ; Check comparator interrupt flag
    LGOTO       ISR_Exit
    MOVIA       0x40
    XORAR       PORTB,1                         ; PB6 toggle output, when compare occur
    BCR         PIR2, CMPIF                     ; Clear comparator interrupt flag
ISR_Exit:
    RETIE
```

12.Analog-to-Digital Convertor (ADC)

12.1 Overview

AT8F2481 provide 22+2+2 channel 12-bit SAR ADC to transfer analog signal into 12-bits digital data. The ADC high reference voltage is selectable. They can be external voltage from PA0, PB1, or internal generated voltage VDD, 4V, 3V or 2V. The Analog input is selected from analog signal input pin PA0~PA7 PB0~PB7, PC0~PC5, from internal generated 1 / 4 *VDD or VSS, and from OPA0 or OPA1. The ADC clock ADCLK can be selected to be F_{INST}/1, F_{INST} /2, F_{INST} /8 or F_{INST} /16. The Sampling pulse width can be selected to be ADCLK*1, ADCLK*2, ADCLK*4 or ADCLK*8. Set ADEN=1 before ADC take into operation. Then set START (ADMD[6]) =1, the ADC will start to convert analog signal to digital.

By the way, user can also by selecting CCP1CON (CCPxM<3:0> = 1011) to set The Special Event Trigger (CCP_ADC_Start), the ADC will start to convert analog signal to digital.

EOC=0 means ADC is in processing. EOC=1 indicate ADC is at end of conversion. If ADIE=1 and global interrupt is enabled, the ADC interrupt will issue after EOC low go high. The block diagram is as following figure36.

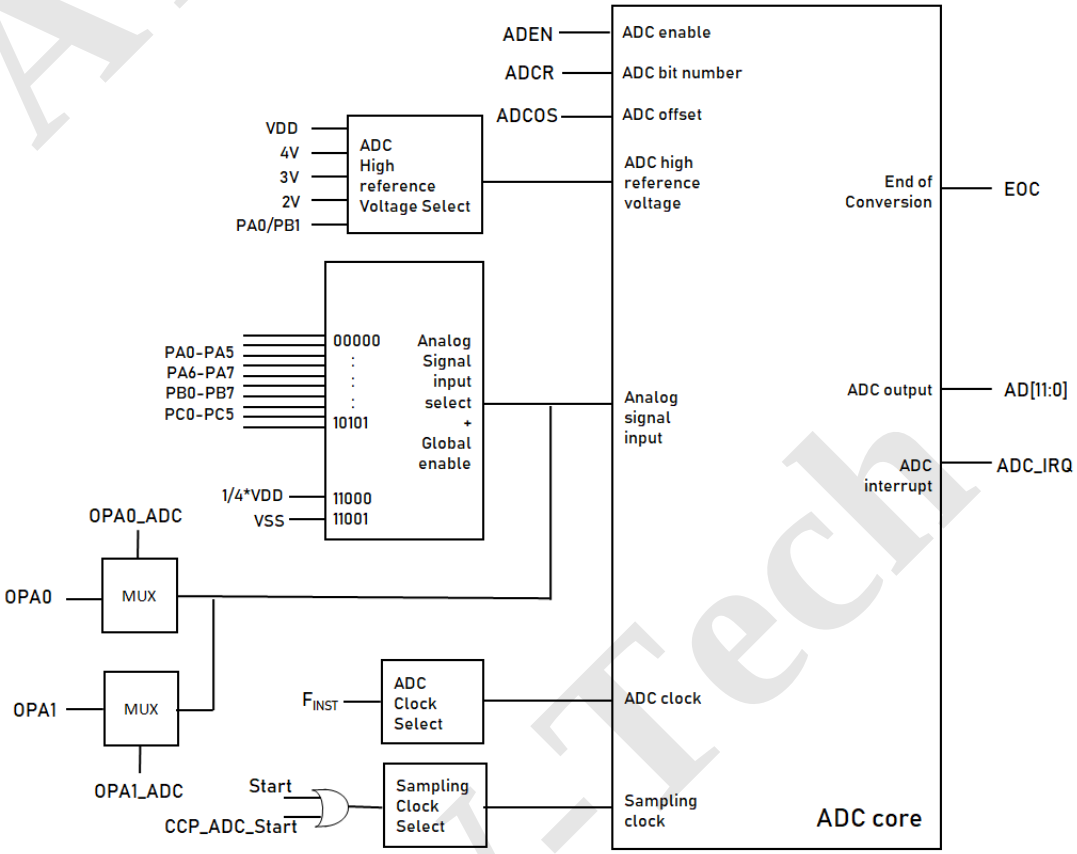


Figure 38 ADC Block Diagram

12.2 ADC Reference Voltage

ADC have built-in five high reference voltage source which is controlled by ADVREFH register (See table 12-1). These high reference voltage source are external voltage source (PA0 or PB1 by option) and four internal voltage source (VDD, 4V, 3V, 2V). When EVHENB bit is “1”, ADC reference voltage is external

voltage source from external voltage source. In this mode PA0 or PB1 must be a voltage between VDD and 2V. If EVHENB bit is 0, ADC reference voltage is from internal voltage source selected by VHS[1:0]. If VHS[1:0] is “11”, ADC reference voltage is VDD. If VHS[1:0] is “10”, ADC reference voltage is 4V. If VHS[1:0] is “01”, ADC reference voltage is 3V. If VHS[1:0] is “00”, ADC reference voltage is 2V. For internal reference voltage of 4V/3V/2V application VDD can't below the selected internal voltage level of 4V, 3V or 2V. ADC input voltage range is from VSS to high reference voltage.

EVHENB	VHS[1:0]	Reference voltage
1	xx	PA0 or PB1 (by option)
0	11	VDD
0	10	4V
0	01	3V
0	00	2V

Table 12-1 ADC Reference Voltage Select

12.3 ADC analog input channel

ADC input pins are shared with digital I/O pins. Connect an analog signal to these pin may cause extra current leakage in I/O pins. In the power down mode, the above leakage current will be a big problem. Write “1” to PxCON register bit will configure related PAx/PBx/PCx (select by PxSEL[1:0]) pin as pure analog input pin to avoid current leakage, and once set it can't be use as normal I/O.

Except setting the PxCON register bit, the selected analog input pin must be set as input mode and the internal pull-high / pull-down must be disabled, otherwise the analog input level may be affected.

ADC use CHS[4:0] to select analog input source.

ADEN	GCHS	CHS[4:0]	ADC analog input source
0	x	xxxxx	x
x	0	xxxxx	x
1	1	00000	PA0
1	1	00001	PA1
1	1	00010	PA2
1	1	00011	PA3
1	1	00100	PA4
1	1	00101	PA5
1	1	00110	PA6
1	1	00111	PA7
1	1	01000	PB0
1	1	01001	PB1
1	1	01010	PB2
1	1	01011	PB3
1	1	01100	PB4
1	1	01101	PB5
1	1	01110	PB6
1	1	01111	PB7
1	1	10000	PC0
1	1	10001	PC1
1	1	10010	PC2
1	1	10011	PC3
1	1	10100	PC4
1	1	10101	PC5
1	1	11100	1 / 4 * VDD
1	1	11101	VSS

Table 12-2 ADC Analog Input Source Select

12.4 ADC clock (ADCLK), sampling clock (SHCLK) and bit number

The conversion speed and accuracy are affected by the selection of the ADC clock (ADCLK), sampling pulse width (SHCLK), and conversion bit number. ADCLK serves as the base clock of the ADC. During the operation of the SAR ADC, each bit decision is synchronized with ADCLK. SHCLK defines the duration of the analog signal sampling period. A longer SHCLK allows the sampled voltage to more closely approach the original analog signal level, but it also reduces the overall conversion speed, and vice versa. The ADC can select different conversion bit number which is depended on ADCR[1:0] register bits. There are 2 bit number to select, which is 12-bit, 10-bit and 8-bit. Less conversion bit number will speed up the ADC conversion rate but the effective ADC bit is less. More conversion bit number will slow down the conversion rate but the accuracy is more.

The ADC clock is derived from F_{INST} and is selectable from ADCK[1:0].

ADCK[1:0]	ADC clock
00	F_{INST}/16
01	F_{INST}/8
10	F_{INST}/1
11	F_{INST}/2

Table 12-3 ADC Clock Select

The Sampling clock width is derived from ADCLK and is selectable from SHCK[1:0].

SHCK[1:0]	Sampling clock
00	1 ADCLK
01	2 ADCLK
10	4 ADCLK
11	8 ADCLK

Table 12-4 ADC Sampling Clock Select

ADC bit number select is from ADCR[1:0].

ADCR[1:0]	Conversion bit number
00	8-bit
01	10-bit
1x	12-bit

Table 12-5 Conversion bit Number Select

The ADC converting time is from START(Start to ADC convert) to EOC=1 (End of ADC convert). The duration is depending on ADC resolution and ADC clock rate and sampling clock width.

ADC conversion time ≈ sampling clock width + (ADC bit number + 2) * ADCLK width.

The following table is some example conversion time and conversion rate of ADC.

Bit No.	ADC clock	SHCLK	Conversion Time (ADCLK No.)	F_{INST} =2MHz		F_{INST} =250KHz	
				Time	Rate	Time	Rate
12	F_{INST}/16	8 ADCLK	22	176us	5.68kHz	1408us	710Hz
12	F_{INST}1	1 ADCLK	15	7.5us	133.3kHz	60us	16.7kHz
10	F_{INST}/1	1 ADCLK	13	6.5us	153.8kHz	52us	19.2kHz
8	F_{INST}/1	1 ADCLK	11	5.5us	181.8kHz	44us	22.7kHz

Table 12-6 ADC Conversion Time

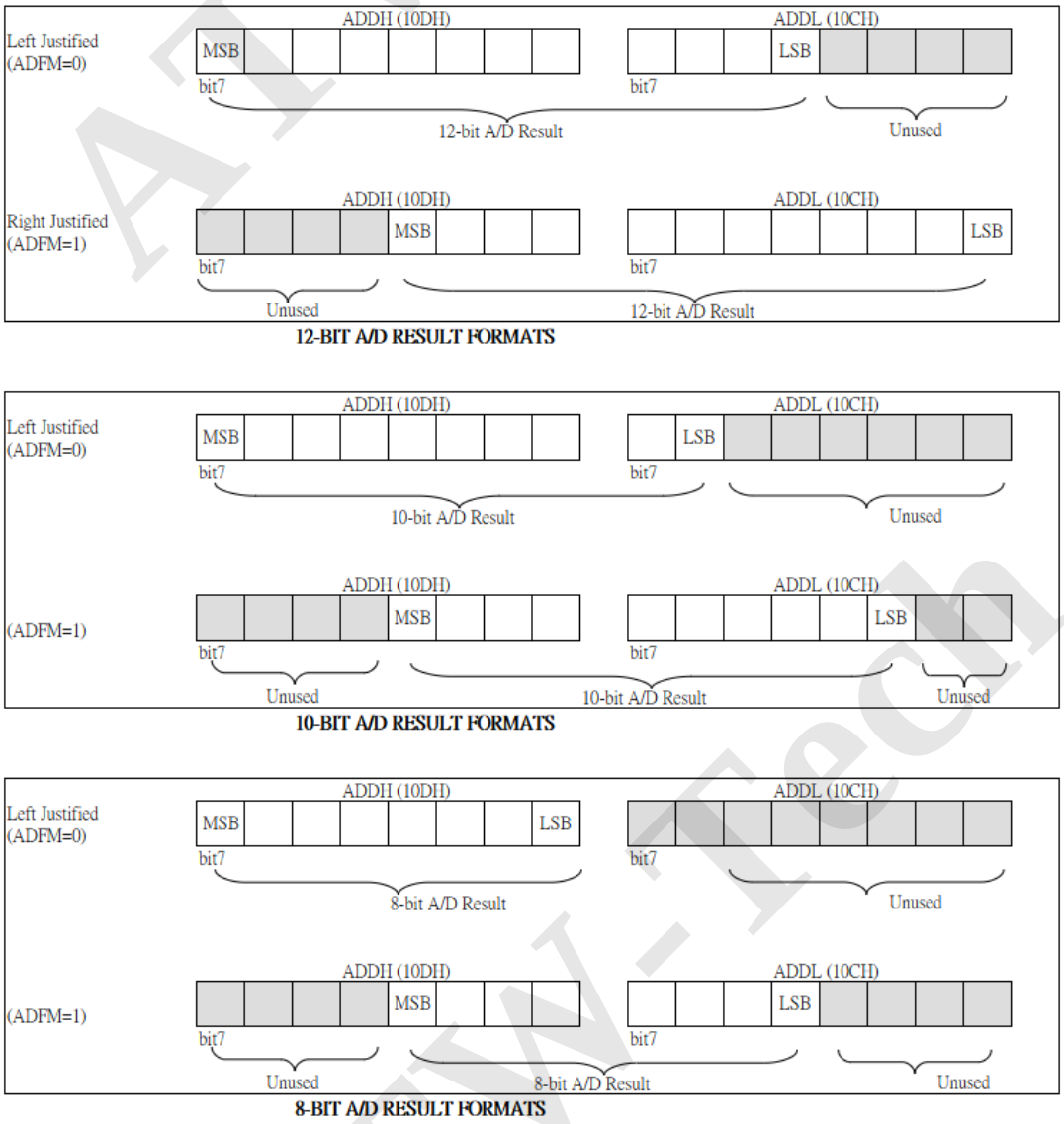
12.5 ADC operation

Set ADC clock (ADCLK), sampling clock width (SHCLK), conversion bit number (ADCR), ADC high reference voltage (ADVREFH), select input channel and PACON or PBCON related bit. Then set ADEN=1.

After setting ADEN=1, it must wait at least 256us (ADC internal bias stable time) before ADC can operate. Write START to 1 to start an ADC conversion session. During ADC is in processing EOC=0. Polling EOC=1 or wait for ADC interrupt at the end of ADC conversion.

12.6 ADC Data Format

ADC 12Bit Data output form [ADDH+ADDL]. User can use ADFM setting. ADC Out data[11:0]. The figure below shows the A/D result data format of AT8F2481 for different conversion bit number



12-BIT A/D RESULT FORMATS

10-BIT A/D RESULT FORMATS

8-BIT A/D RESULT FORMATS

Figure 39 ADFM Setting

12.6 ADC Control Register

12.6.1 ADMD (ADC mode Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADMD	0x109	ADEN	START	GCHS	CHS4	CHS3	CHS2	CHS1	CHS0
R/W Property		R/W	W	R	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7 **ADEN:** ADC enable bit
1 = ADC is enabled
0 = ADC is disabled

Bit 6 **START:** Start an ADC conversion session. Write 1 to this bit will start ADC converting. This bit is write-only. Read this bit will get 0.

Bit 5 **GCHS:** ADC global channel select bit
1 = Enable all ADC input channel
0 = Disable all ADC channel

Bit 4:0 **CHS[4:0]:** Comparator output inverse control bit
00000 = Select PA0 pad as ADC input
00001 = Select PA1 pad as ADC input
00010 = Select PA2 pad as ADC input
00011 = Select PA3 pad as ADC input
00100 = Select PA4 pad as ADC input
00101 = Select PA5 pad as ADC input
00110 = Select PA6 pad as ADC input
00111 = Select PA7 pad as ADC input
01000 = Select PB0 pad as ADC input
01001 = Select PB1 pad as ADC input
01010 = Select PB2 pad as ADC input
01011 = Select PB3 pad as ADC input
01100 = Select PB4 pad as ADC input
01101 = Select PB5 pad as ADC input
01110 = Select PB6 pad as ADC input
01111 = Select PB7 pad as ADC input
10000 = Select PC0 pad as ADC input
10001 = Select PC1 pad as ADC input
10010 = Select PC2 pad as ADC input
10011 = Select PC3 pad as ADC input
10100 = Select PC4 pad as ADC input
10101 = Select PC5 pad as ADC input
11100 = Select 1/4 VDD as ADC input
11101 = Select VSS as ADC input

12.6.2 ADDL (ADC LSB output Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDL	0x10C	ADDL[7:0]							
R/W Property		R							
Initial Value		xxxxxxxx							

Bit 7:0 **ADDL[7:0]:** Low-byte ADC data buffer

12.6.3 ADDH (ADC output data Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDH	0x10D	ADDH[7:0]							
R/W Property		R							
Initial Value		xxxxxxxx							

Bit 7:0 **ADDH[7:0]:** High-byte ADC data buffer

12.6.4 ADCON1 (ADC Control Register 1)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	0x10E	EVHENB	-	EOC	ADFM	ADCK1	ADCK0	VHS1	VHS0
R/W Property		R/W	-	R	R/W	R/W	R/W	R/W	R/W
Initial Value		0	x	1	0	0	0	1	1

Bit 7 **EVHENB:** ADC reference high voltage (VREFH) select control bit
1 = ADC reference high voltage is supplied by external pin PA0
0 = ADC reference high voltage is internal generated, the voltage selected depends on VHS1~0

Bit 6 **Unimplemented**

Bit 5 **EOC:** ADC status bit, read-only
1 = ADC is end-of-convert, the ADC data present in ADR and ADD is available
0 = ADC is in procession

Bit 4 **ADFM:** ADC Result Format Select bit
1 = Right justified
0 = Left justified

Bit 3:2 **ADCK[1:0]:** ADC clock select
11 = ADC clock=F_{INST}/2
10 = ADC clock=F_{INST}/1
01 = ADC clock=F_{INST}/8
00 = ADC clock=F_{INST}/16

Bit 1:0 **VHS[1:0]:** ADC internal reference high voltage select bits
11 = VREFH=VDD
10 = VREFH=4V
01 = VREFH=3V,
00 = VREFH=2V

12.6.5 ADJMD (ADC analog pin Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADJMD	0x10F	-	-	ADJ_SIGN	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0
R/W Property		-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		x	x	0	0	0	0	0	0

Bit 7:6 **Unimplemented**

Bit 5 **ADJ_SIGN:** ADC adjust sign bit
1 = ADC data increase
0 = ADC data decrease

Bit 4:0 **ADJ[4:0]:** ADC adjust offset
11111 = Offset 16mV
00000 = Offset 0mV

12.6.6 ADCR (Sampling pulse and ADC bit Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCR	0x111	PxSEL1	PxSEL0	-	-	SHCK1	SHCK0	ADCR1	ADCR0
R/W Property		R/W	R/W	-	-	R/W	R/W	R/W	R/W
Initial Value		0	0	x	x	1	0	1	0

Bit 7:6 **PxSEL[1:0]:** Select bits for mapping of virtual SFR PxCON register to PACON, PBCON or PCCON.
10 = PORTC analog pin setting (PCCON) via virtual SFR PxCON
01 = PORTB analog pin setting (PBCON) via virtual SFR PxCON
00 = PORTA analog pin setting (PACON) via virtual SFR PxCON

Bit 5:4 **Unimplemented**

Bit 3:2 **SHCK[1:0]:** Sampling pulse width select
11 = 8 ADC clock
10 = 4 ADC clock
01 = 2 ADC clock,
00 = 1 ADC clock

Bit 1:0 **ADCR[1:0]:** ADC conversion bit no. select
1x = 12-bit ADC.
01 = 10-bit ADC.
00 = 8-bit ADC.

13.Watch-Dog Timer (WDT)

13.1 Overview

There is an on-chip free-running oscillator in AT8F2481 which is used by WDT. As this oscillator is independent of other oscillation circuits, WDT may still keep working during Standby mode and Halt mode.

WDT can be enabled or disabled by a configuration word. When WDT is enabled by configuration word, its operation still can be controlled by register bit WDTEN (PCON[7]) during program execution. Moreover, the mechanism after WDT time-out can reset AT8F2481 or issue an interrupt request which is determined by another configuration word. At the same time, register bit /TO (STATUS[4]) will be clear to 0 after WDT time-out.

The baseline of WDT time-out period can be 3.5ms, 15ms, 60ms or 250ms which is determined by two configuration words. The time-out period can be lengthened if Prescaler0 is assigned to WDT. Prescaler0 will be assigned to WDT by writing 1 to register bit PS0WDT. The dividing rate of Prescaler0 for WDT is determined by register bits PS0SEL[2:0] and depends on WDT time-out mechanism. The dividing rate is from 1:1 to 1:128 if WDT time-out will reset AT8F2481 and dividing rate is from 1:2 to 1:256 if WDT time-out will interrupt AT8F2481.

When Prescaler0 is assigned to WDT, the execution of instruction CLRWDT will clear WDT, Prescaler0 and set /TO flag to 1.

If user selects interrupt for WDT time-out mechanism, register bit WDTIF (PIR1[5]) will set to 1 after WDT is expired. It may generate an interrupt request if register bit WDTIE (PIE1[5]) and GIE both set to 1. WDTIF will not be clear until firmware writes 0 to WDTIF.

14.Interrupt

14.1 Overview

AT8F2481 provides 14 hardware interrupts:

- Timer0 overflow interrupt.
- Timer1 underflow interrupt.
- Timer4 underflow interrupt.
- Timer5 underflow / CCP interrupt.
- WDT timeout interrupt.
- PA/PB/PC input change interrupt.
- External 0 interrupt.
- External 1 interrupt
- External 2 interrupt
- LVD interrupt.
- Comparator output status change interrupt.
- ADC end-of-convert interrupt.
- SIM interrupt. (Serial Interface Mode interrupt)
- End of EEPROM write interrupt.

GIE is global interrupt enable flag. It has to be 1 to enable hardware interrupt functions.

At the same time, GIE will be clear to 0 by AT8F2481 automatically. The last instruction of interrupt service routine of software interrupt has to be RETIE. Execution of this instruction will set GIE to 1 and return to original execution sequence.

While any of hardware interrupts is occurred, the corresponding bit of interrupt flag will be set to 1. This bit will not be clear until firmware writes 0 to this bit. Therefore, user can obtain information of which event causes hardware interrupt by polling the corresponding bit of interrupt flag. Note that only when the corresponding interrupt enable bit is set to 1, will the corresponding interrupt flag be read. And if the corresponding interrupt enable bit is set to 1 and GIE is also 1, hardware interrupt will occur and next instruction will be fetched from 0x004. At the same time, the register bit GIE will be clear by AT8F2481 automatically.

14.1.1 Timer0 Overflow Interrupt

Timer0 overflow (from 0x00 to 0xFF) will set register bit T0IF. This interrupt request will be serviced if T0IE and GIE are set to 1.

14.1.2 Timer1 Underflow Interrupt

Timer1 underflow (from 0x3FF to 0x00) will set register bit T1IF. This interrupt request will be serviced if T1IE, PEIE and GIE are set to 1.

14.1.3 Timer4 Underflow Interrupt

Timer4 underflow (from 0x3FF to 0x00) will set register bit T4IF. This interrupt request will be serviced if T4IE, PEIE and GIE are set to 1.

14.1.4 Timer5 Underflow / CCP Interrupt

When CCP compare mode and CCP capture mode are both disabled, Timer5 underflow (from 0x3FF to 0x00) will set register bit T5IF/CCPIF. When CCP capture mode is enabled and a capture is made, the T5IF/CCPIF is set. When CCP compare mode is enabled and a comparison is matched, the T5IF/CCPIF is set. When T5IF/CCPIF is set, this interrupt request will be serviced if T5IE/CCPIE, PEIE and GIE are all set to 1.

14.1.5 WDT Timeout Interrupt

When WDT is timeout and the configuration word selects WDT timeout will generate interrupt request, it will set register bit WDTIF. This interrupt request will be serviced if WDTIE, PEIE and GIE are set to 1.

14.1.6 PA/PB/PC Input Change Interrupt

When PAx, $0 \le x \le 7$, PBy, $0 \le y \le 7$, PCz, $0 \le z \le 5$ is configured as input pin and corresponding register bit WUPAx, WUPBx is set to 1, a level change on these selected I/O pin(s) will set register bit PABCIF. This interrupt request will be serviced if PABCIE and GIE are set to 1. Note when PA3, PA4 or PA5 is both set as level change interrupt and external interrupt, the external interrupt enables EIS0, EIS1 or EIS2=1 will disable PA3, PA4 or PA5 level change operation.

14.1.7 External 0 Interrupt

According to the configuration of EIS0=1 and INTEDG, the selected active edge on I/O pin PA4 will set register bit INT0IF and this interrupt request will be served if INT0IE and GIE are set to 1.

14.1.8 External 1 Interrupt

According to the configuration of EIS1=1 and INTEDG, the selected active edge on I/O pin PA3 will set register bit INT1IF and this interrupt request will be served if INT1IE, PEIE and GIE are set to 1.

14.1.9 External 2 Interrupt

According to the configuration of EIS2=1 and INTEDG, the selected active edge on I/O pin PA5 will set register bit INT2IF and this interrupt request will be served if INT2IE, PEIE and GIE are set to 1.

14.1.10 LVD Interrupt

When VDD level falls below LVD voltage, LVD flag will from high to low, and set the register bit LVDIF=1. This interrupt request will be serviced if LVDIE, PEIE and GIE are set to 1.

14.1.11 Comparator Output Status Change Interrupt

The comparator interrupt is triggered whenever a change occurs on the comparator output status. This interrupt request will be serviced if CMPIE, PEIE and GIE are set to 1. Note that before the comparator interrupt could happen, reading register CMPCON is needed to clear the previous comparator output status difference.

14.1.12 ADC End of Conversion Interrupt

The ADC interrupt is triggered whenever an ADC end-of-convert signal is issued. This interrupt request will be serviced if ADIE, PEIE and GIE are set to 1.

14.1.13 Serial Interface Mode Interrupt

The SIM interrupt is triggered whenever an SPIF of SPI mode or MIF of I²C mode is issued. This interrupt request will be serviced if SIMIE and GIE are set to 1.

14.2 Interrupt Control Register

14.2.1 INTCON (Interrupt Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCON	0xB	GIE	PEIE	T0IE	INT0IE	PABCIE	T0IF	INT0IF	PABCIF
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7 **GIE:** Global Interrupt enable bit
1 = Enable all interrupts
0 = Disable all interrupts

Bit 6 **PEIE:** Peripheral Interrupt enable bit. Peripheral interrupts are defined to be all interrupts except timer0 interrupt, external 0 interrupt and pad interrupts.
1 = Enable all peripheral interrupts
0 = Disable all peripheral interrupts

Bit 5 **T0IE:** Timer0 Overflow Interrupt enable bit
1 = Enable TMR0 interrupt
0 = Disable TMR0 interrupt

Bit 4 **INT0IE:** External Interrupt 0 enable bit
1 = Enable external interrupt 0
0 = Disable external interrupt 0

Bit 3 **PABCIE:** PORTA/PORTB/PORTC change Interrupt enable bit
1 = Enable PORTA/PORTB/PORTC change interrupts
0 = Disable PORTA/PORTB/PORTC change interrupts

Bit 2 **T0IF:** Timer0 interrupt flag bit
1 = TMR0 has overflowed
0 = TMR0 did not overflow

Bit 1 **INT0IF:** External interrupt 0 flag bit
1 = External interrupt 0 occurred
0 = External interrupt 0 did not occur

Bit 0 **PABCIF:** PORTA/PORTB/PORTC change interrupt flag bit
1 = PORTA/PORTB/PORTC change has changed state
0 = PORTA/PORTB/PORTC change did not change state

14.2.2 PIR1 (Interrupt Flag Register 1)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR1	0xC	INT2IF	INT1IF	WDTIF	-	EEIF	T5IF/ CCPIF	T4IF	T1IF
R/W Property		R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial Value(note*)		0	0	0	x	0	0	0	0

Bit 7 **INT2IF:** External interrupt 2 flag bit
1 = External interrupt 2 is occurred (INT2IF must be clear by firmware)
0 = External interrupt 2 did not occur

Bit 6 **INT1IF:** External interrupt 1 flag bit
1 = External interrupt 1 is occurred (INT1IF must be clear by firmware)
0 = External interrupt 1 did not occur

Bit 5 **WDTIF:** WDT timeout interrupt flag bit
1 = WDT timeout interrupt is occurred (WDTIF must be clear by firmware)
0 = WDT timeout did not occur

Bit 4 **Unimplemented**

Bit 3 **EEIF:** End of EEPROM writing interrupt flag bit
1 = End of EEPROM writing interrupt is occurred (EEIF must be clear by firmware)
0 = EEPROM interrupt did not occur

Bit 2 **T5IF/CCPIF:** Timer5 or CCP interrupt flag
1 = Timer5 or CCP interrupt is occurred (T5IF/CCPIF must be clear by firmware)
0 = Timer5 or CCP interrupt did not occur

Bit 1 **T4IF:** Timer4 interrupt flag
1 = Timer4 interrupt is occurred (T4IF must be clear by firmware)
0 = Timer4 interrupt did not occur

Bit 0 **T1IF:** Timer1 interrupt flag
1 = Timer1 interrupt is occurred (T1IF must be clear by firmware)
0 = Timer1 interrupt did not occur

Note: When corresponding PIE1 bit is not enabled, the read interrupt flag is 0.

14.2.3 PIE1 (Interrupt Enable Register 1)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIE1	0x8C	INT2IE	INT1IE	WDTIE	-	EEIE	T5IE/CCPIE	T4IE	T1IE
R/W Property		R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial Value(note*)		0	0	0	x	0	0	0	0

Bit 7 **INT2IE:** External interrupt 2 enable bit
1 = External interrupt 2 is enable
0 = External interrupt 2 is disable

Bit 6 **INT1IE:** External interrupt 1 enable bit
1 = External interrupt 1 is enable
0 = External interrupt 1 is disable

Bit 5 **WDTIF:** WDT timeout interrupt enable bit
1 = WDT timeout interrupt is enable
0 = WDT timeout interrupt is disable

Bit 4 **Unimplemented**

Bit 3 **EEIE:** End of EEPROM Writing interrupt enable bit
1 = Enable End of EEPROM Writing interrupt
0 = Disable End of EEPROM Writing interrupt

Bit 2 **T5IE/CCPIE:** Timer5 or CCP interrupt enable bit
1 = Timer5 or CCP interrupt is enable
0 = Timer5 or CCP interrupt is disable

Bit 1 **T4IE:** Timer4 underflow interrupt enable bit
1 = Timer4 underflow interrupt is enable
0 = Timer4 underflow interrupt is disable

Bit 0 **T1IF:** Timer1 underflow interrupt enable bit

1 = Timer1 underflow interrupt is enable

0 = Timer1 underflow interrupt is disable

14.2.4 PIR2 (Interrupt Flag Register 2)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR2	0xD	ADIF	LVDIF	CMPIF	-	SIMIF	LSRIF	TXIF	RXIF
R/W Property		R/W	R/W	R/W		R	R	R	R
Initial Value (note*)		0	0	0	x	0	0	1	0

Bit 7 **ADIF:** ADC end of conversion interrupt flag bit.
1 = ADC end of conversion interrupt is occurred (ADIF must be clear by firmware)
0 = ADC interrupt did not occur

Bit 6 **LVDIF:** LVD interrupt flag bit
1 = LVD interrupt is occurred (LVDIF must be clear by firmware)
0 = LVD interrupt did not occur

Bit 5 **CMPIF:** Comparator **output change state** interrupt flag
1 = Comparator interrupt is occurred (CMPIF must be clear by firmware)
0 = Comparator interrupt did not occur

Bit 4 **Unimplemented**

Bit 3 **SIMIF:** Serial port interrupt flag bit (I^2C mode or SPI mode)
1 = Serial port interrupt is occurred (SIMIF must be clear by firmware at MIF or SPIF)
0 = Serial port interrupt is not occurred
If SIMCR.MEN=1. SIMIF show status of MIF
If SIMCR.SPE=1. SIMIF show status of SPIF

Bit 2 **LSRIF:** LSR interrupt flag bit
1 = Line Status interrupt is occurred (LSRIF must be clear by read LSR)
(BKINT=1 or FERR=1 or PERR=1 or OVERR=1)
0 = Line Status interrupt did not occur

Bit 1 **TXIF:** Transmitter holding register (THR) empty flag
TXIF show status of LSR.
This bit indicates the controller is ready to accept a new character for transmission. It is set to logic 1 when a character is transferred from the transmitter holding register (TBR) into the transmitter shift register. Write data to TBR will clear this flag.

Bit 0 **RXIF:** Data ready flag bit
RXIF show status of LSR.RREADY
It is set to logic 1 whenever a complete incoming character has been received and transferred into the receiver buffer register. Read the RBR data will clear this flag.

Note: When corresponding PIE2 bit is not enabled, the read interrupt flag (ADIF, LVDIF, CMPIF) is 0.

14.2.5 PIE2 (Interrupt Enable Register 2)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIE2	0x8D	ADIE	LVDIE	CMPIE	-	SIMIE	LSRIE	TXIE	RXIE
R/W Property		R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial Value(note*)		0	0	0	x	0	0	0	0

Bit 7 **ADIE:** ADC end of conversion interrupt enable bit
1 = ADC end of conversion interrupt is enable
0 = ADC end of conversion interrupt is disable

Bit 6 **LVDIE:** LVD interrupt enable bit
1 = LVD interrupt is enable
0 = LVD interrupt is disable

Bit 5 **CMPIE:** Comparator interrupt enable bit
1 = Comparator interrupt is enable
0 = Comparator interrupt is disable

Bit 4 **Unimplemented**

Bit 3 **SIMIE:** Serial port interrupt enable bit. (I^2C mode or SPI)
1 = Serial port interrupt is enable. (I^2C mode or SPI mode)
If SIMCR.MEN=1. I^2C interrupt is enable
If SIMCR.SPE=1. SPI interrupt is enable
0 = Serial port interrupt is disable

Bit 2 **LSRIE:** LSR interrupt enable bit
1 = Receiver line status interrupt is enable
0 = Receiver line status interrupt is disable

Bit 1 **TXIE:** Transmit holding register (THR) empty interrupt enable bit
1 = Transmit holding register (THR) empty interrupt is enable
0 = Transmit holding register (THR) empty interrupt is disable

Bit 0 **RXIE:** Receive one byte completely interrupt enable bit
1 = Receive one byte completely interrupt is enable
0 = Receive one byte completely interrupt is disable

14.2.6 PCON (Power Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	0x8F	WDTEN	-	LVDEN	-	LVREN	-	-	-
R/W Property		R/W	-	R/W	-	R/W	-	-	-
Initial Value		1	x	0	x	1	x	x	x

Bit 7 **WDTEN:** Enable/disable WDT
1 = Enable WDT
0 = Disable WDT

Bit 6 **Unimplemented**

Bit 5 **LVDEN:** Enable/disable LVD
1 = Enable LVD
0 = Disable LVD

Bit 4 **Unimplemented**

Bit 3 **LVREN:** Enable/disable LVR
1 = Enable LVR
0 = Disable LVR

Bit 2:0 **Unimplemented**

14.2.7 INTEDG (Interrupt Edge Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTEDG	0x9B	INT2EDG	EIS2	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	1	0	1

Bit 7 **INT2EDGE:** INT2 edge trigger select bit

1 = Rising edge

0 = Falling edge

Bit 6 **EIS2:** External interrupt 2 select bit

1 = PA5 is external interrupt 2

0 = PA5 is GPIO

Bit 5 **EIS1:** External interrupt 1 select bit

1 = PB1 or PA3 is external interrupt 1

0 = PB1 or PA3 is GPIO

Bit 4 **EIS0:** External interrupt 0 select bit

1 = PB0, PB4 or PA4 is external interrupt 0

0 = PB0, PB4 or PA4 is GPIO

Bit 3:2 **INT1G[1:0]:** INT1 edge trigger select bit

11 = Rising / falling edge

10 = Falling edge

01 = Rising edge

00 = reserved

Bit 1:0 **INT0G[1:0]:** INT0 edge trigger select bit

11 = Rising / falling edge

10 = Falling edge

01 = Rising edge

00 = reserved

15.Oscillation Configuration

15.1 Overview

Because AT8F2481 is a dual-clock IC, there are high oscillation (F_{HOSC}) and low oscillation (F_{LOSC}) that can be selected as system oscillation (F_{OSC}). The oscillators which could be used as F_{HOSC} are internal high RC oscillator (I_HRC), external high crystal oscillator (E_HXT) and external crystal oscillator (E_XT). The oscillators which could be used as F_{LOSC} are internal low RC oscillator (I_LRC) and external low crystal oscillator (E_LXT). When external crystal (E_HXT, E_XT or E_LXT) is adopted for high oscillation or low oscillation according to setting of configuration words, PA7 will be used as crystal input pin (Xin) and PA6 will be used as crystal output pin (Xout). When I_HRC or I_LRC mode is selected as system oscillation and E_HXT, E_XT or E_LXT is not adopted, instruction clock is observable on PA7 if a configuration word is enabled.

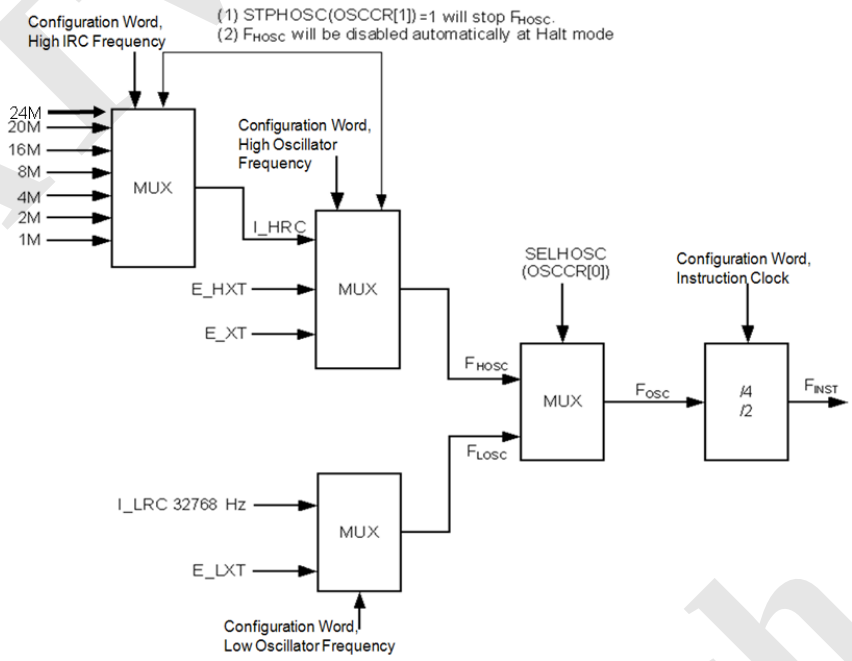


Figure 40 Oscillation Configuration of AT8F2481

There are two configuration words to determine which oscillator will be used as F_{HOSC}. When I_HRC is selected as F_{HOSC}, I_HRC output frequency is determined by three configuration words and it can be 1M, 2M, 4M, 8M, 16M ,20MHZ or 24 MHz. Moreover, external crystal oscillator pads PA6 and PA7 can be used as I/O pins. On the other hand, PA7 can be the output pin of instruction clock according to a configuration word's setting. If F_{HOSC} required external crystal whose frequency ranges from 8MHz to 24 MHz, E_HXT is recommended. If F_{HOSC} required external crystal whose frequency ranges from 455KHz to 6MHz, E_XT is recommended. When E_HXT or E_XT is adopted, PA6/PA7 cannot be used as I/O pins. They must be used as crystal output pin and input pin. PA7 is crystal output pin (Xout) and PA6 is crystal input pin (Xin).

There is one configuration word to determine which oscillator will be used as F_{LOSC}. When I_LRC is selected, its frequency is centered on 32768Hz. If F_{LOSC} required external crystal, E_LXT is selected and only 32768Hz crystal is allowed. When E_LXT is adopted, PA6/PA7 cannot be used as I/O pins. They must be used as crystal output pin and input pin. PA7 is crystal output pin (Xout) and PA6 is crystal input pin (Xin). The dual-clock combinations of F_{HOSC} and F_{LOSC} are listed below.

No.	F_{HOSC}	F_{LOSC}
1	I_HRC	I_LRC
2	E_HXT or E_XT	I_LRC
3	I_HRC	E_LXT

Table 15-1 Dual-Clock Combinations

When E_HXT, E_XT or E_LXT is used as one of oscillations, the crystal or resonator is connected to Xin and Xout to provide oscillation. Moreover, a resistor and two capacitors are recommended to connect as following figure in order to provide reliable oscillation, refer to the specification of crystal or resonator to adopt appropriate C1 or C2 value. The recommended value of C1 and C2 are listed in the table below.

Oscillation Mode	Crystal Frequency (Hz)	C1, C2 (pF)
E_HXT	24M	5 ~ 10
	20M	5 ~ 10
	16M	5 ~ 10
	10M	5 ~ 30
	8M	5 ~ 20
E_XT	4M	5 ~ 30
	1M	5 ~ 30
	455K	10 ~ 100
E_LXT	32768	5 ~ 30

Table 15-2 Recommended C1 and C2 Value for Different Kinds of Crystal Oscillation

For 20MHZ resonator in 2 clock CPU cycle mode, an 18pF C2 capacitor is a must.

To get precise and stable 32.768k frequency, choosing the right C1 and C2 value is important. You need to match the C1 / C2 capacitance to the specific crystal you chose. Every crystal datasheet lists something called the Load Capacitance (CL), C1 and C2 value is chosen with the following formula:

C1=C2=2*CL-Cbt

Where Cbt is the AT8F2481 crystal pad built-in capacitance, which is about 5pF. For example, for crystal CL=12.5P, C1=C2=20pF is recommended.

The accuracy of I_HRC is ±1% at 25°C commercial conditions.

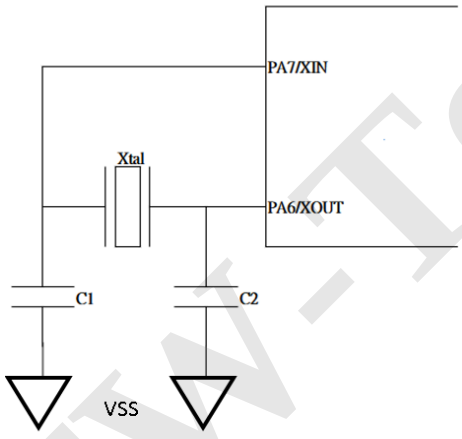


Figure 41 Connection for External Crystal Oscillation

Either F_{HOSC} or F_{LOSC} can be selected as system oscillation F_{OSC} according to the value of register bit SELHOSC (OSCCR[0]). When SELHOSC is 1, F_{HOSC} is selected as F_{OSC}. When SELHOSC is 0, F_{LOSC} is selected as F_{OSC}. Once F_{OSC} is determined, the instruction clock F_{INST} can be $F_{OSC}/2$ or $F_{OSC}/4$ according to value of a configuration word.

15.2 OSCCR (Oscillation Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	0x9	-	-	-	XSPD_STP	OPMD[1:0]		STPHOSC	SELHOSC
R/W Property		-	-	-	W	R/W		R/W	R/W
Initial Value		x	x	x	0	00		0	1

Bit 7:5 **Unimplemented**

Bit 4 **XSPD_STP:** Write 1 to stop crystal 32.768K speed-up function, write-only

Bit 3:2 **OPMD[1:0]:** Selection of operating mode

OPMD[1:0]	Operating Mode
00	Normal mode
01	Halt mode
10	Standby mode
11	reserved

Bit 1 **STPHOSC:** Disable/enable high-frequency oscillation (F_{HOSC})

1 = F_{HOSC} will stop oscillation and be disabled

0 = F_{HOSC} keep oscillation

Bit 0 **SELHOSC:** Selection of system oscillation (F_{OSC})

1 = F_{OSC} is high-frequency oscillation (F_{HOSC})

0 = F_{OSC} is low-frequency oscillation (F_{LOSC})

Note: STPHOSC cannot be changed with SELHOSC or OPMD at the same time. STPHOSC cannot be changed with OPMD at the same time during SELHOSC=1.

15.3 OSCCAL (Internal Oscillator Calibration)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCALL	0x18E	OSC CAL7	OSC CAL6	OSC CAL5	OSC CAL4	OSC CAL3	OSC CAL2	OSC CAL1	OSC CAL0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		F_{V_HRC} 8 bits option trim data bit[7:0]							

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCALH	0x18F	-	-	-	-	-	-	-	OSC CAL8
R/W Property		-	-	-	-	-	-	-	R/W
Initial Value		x	x	x	x	x	x	x	F_{V_HRC} 8 bits option trim data bit[8]

Bit 8:0 **OSCCAL[8:0]:** F_{V_HRC} 9 bits calibration data load to OSCCAL8~OSCCAL0 and usercan adjust OSCCAL8~0 at slow mode. Note OSCCALH Bit[0] of F_{V_HRC} 8 bits option trim data MSB bit[8].

16. Operating Mode

16.1 Overview

AT8F2481 provides four kinds of operating mode to tailor all kinds of application and save power consumptions. These operating modes are Normal mode, Slow mode, Standby mode and Halt mode. Normal mode is designated for high-speed operating mode. Slow mode is designated for low-speed mode in order to save power consumption. At Standby mode, AT8F2481 will stop almost all operations except Timer0/Timer1/Timer4/Timer5/WDT in order to wake-up periodically. At Halt mode, AT8F2481 will sleep until external event or WDT trigger IC to wake-up. The block diagram of four operating modes is described in the following figure.

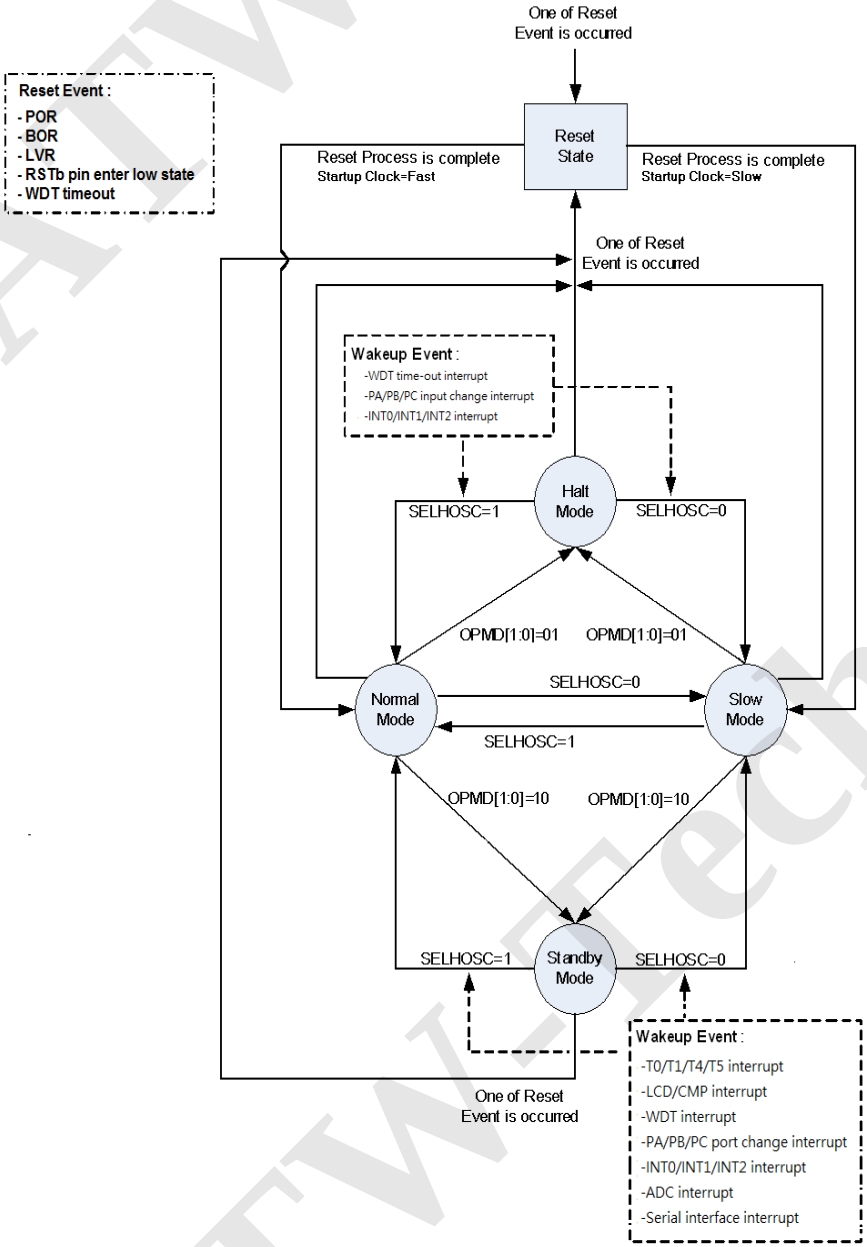


Figure 42 Four Operating Modes

16.2 Normal Mode

After any Reset Event is occurred and Reset Process is completed, AT8F2481 will begin to execute program under Normal mode or Slow mode. Which mode is selected after Reset Process is determined by the Startup Clock configuration word. If Startup Clock=fast, AT8F2481 will enter Normal mode, if Startup Clock=Slow, AT8F2481 will enter Slow mode. At Normal mode, F_{HOSC} is selected as system oscillation in order to provide highest performance and its power consumption will be the largest among four operating modes. After power on or any reset trigger is released, AT8F2481 will enter Normal mode after reset process is completed.

- Instruction execution is based on F_{HOSC} and all peripheral modules may be active according to
- corresponding module enable bit.
- The F_{LOSC} is still active and running.
- IC can switch to Slow mode by writing 0 to register bit SELHOSC (OSCCR[0]).
- IC can switch to Standby mode or Halt mode by programming register bits OPMD[1:0] (OSCCR[3:2]).
- For real time clock applications, the AT8F2481 can run in normal mode, at the same time the low-frequency clock Low Oscillator Frequency connects to Timer0 clock. This is made possible by setting LCKTM0 to 1 and corresponding configuration word Timer0 source setting to 1.

16.3 Slow Mode

AT8F2481 will enter Slow mode by writing 0 to register bit SELHOSC. At Slow mode, F_{LOSC} is selected as system oscillation in order to save power consumption but still keep IC running. However, F_{HOSC} will not be disabled automatically by AT8F2481. Therefore user can write 0 to register bit STPHOSC (OSCCR[1]) in slow mode to reduce power consumption further. But it is noted that it is forbidden to enter slow mode and stop F_{HOSC} at the same time, one must enter slow mode first, then disable F_{HOSC}, or the program may hang on.

- Instruction execution is based on F_{LOSC} and all peripheral modules may be active according to corresponding module enable bit.
- F_{HOSC} can be disabled by writing 1 to register bit STPHOSC.
- IC can switch to Standby mode or Halt mode by programming register bits OPMD[1:0].
- IC can switch to Normal mode by writing 1 to SELHOSC.

16.4 Standby Mode

AT8F2481 will enter Standby mode by writing 10b to register bits OPMD[1:0]. At Standby mode, however, F_{HOSC} will not be disabled automatically by AT8F2481 and user has to enter slow mode and write 1 to register bit STPHOSC in order to stop F_{HOSC} oscillation. Most of AT8F2481 peripheral modules are disabled but Timer can be still active if register bit T0EN/T1EN/T4EN/T5EN is set to 1. Therefore, AT8F2481 can wake-up after Timer0/Timer1/Timer4/Timer5 is expired. The expiration period is determined by the register TMR0/TMR1[9:0]/TMR4[9:0]/TMR5[9:0], F_{INST} and other configurations for Timer0/Timer1/Timer4/Timer5.

- Instruction execution is stop and some peripheral modules may be active according to corresponding module enable bit.
- F_{HOSC} can be disabled by writing 1 to register bit STPHOSC.
- The F_{LOSC} is still active and running.
- IC can wake-up from Standby mode if any of (a) Timer0/Timer1/Timer4/Timer5 (overflow/underflow) interrupt, (b) WDT timeout interrupt, (c) PA/PB/PC input change interrupt, (d) INT external interrupt is

happened, (e) LVD interrupt, (f) Comparator output status change interrupt or (g) ADC end-of-convert interrupt.

- After wake-up from Standby mode, IC will return to Normal mode if SELHOSC=1, IC will return to Slow mode if SELHOSC=0.
- It is not recommended to change oscillator mode (normal to slow / slow to normal) and enter standby mode at the same time.

The user should add a NOP instruction immediately after Standby Mode instruction.

```
; Into Standby Mode Method:
MOVIA   0x08 | C_FHOSC_Sel
MOVAR   OSCCR               ; Set OSCCR Register to Enters Standby Mode
NOP                         ; NOP Instruction After the Standby Mode
```

16.5 Halt Mode

AT8F2481 will enter Halt mode by executing instruction SLEEP or writing 01b to register bits OPMD[1:0]. After entering Halt mode, register bit /PD (STATUS[3]) will be clear to 0, register bit /TO (STATUS[4]) will be set to 1 and WDT will be clear but keep running.

At Halt mode, all of peripheral modules are disabled, instruction execution is stop and AT8F2481 can only wake-up by some specific events. Therefore, Halt mode is the most power saving mode provided by AT8F2481.

- Instruction execution is stop and all peripheral modules are disabled.
- F_{HOSC} and F_{LOSC} are both disabled automatically.
- IC can wake-up from Halt mode if any of (a) WDT timeout interrupt, (b) PA/PB/PC input change interrupt or (c) INT or external interrupt is happened.
- After wake-up from Halt mode, IC will return to Normal mode if SELHOSC=1, IC will return to Slow mode if SELHOSC=0.

Note: Users can change STPHOSC and enter Halt mode in the same instruction.

- It is not recommended to change oscillator mode (normal to slow or slow to normal) and enter halt mode at the same time.

16.6 Wake-up Stable Time

The wake-up stable time of Halt mode is determined by Configuration word: High Oscillator Frequency or Low Oscillator Frequency. If one of E_HXT, E_XT and E_LXT is selected, the wake-up period would be 512*F_{OSC}. And if no XT mode are selected, 16*Fosc would be set as wake up period. On the other hand, there is no need of wake-up stable time for Standby mode because either F_{HOSC} or F_{LOSC} is still running at Standby mode.

Before AT8F2481 enter Standby mode or Halt mode, user may execute instruction BSR INTCON.GIE At this condition, AT8F2481 will branch to address 0x004 in order to execute interrupt service routine after wake-up. If instruction BCR INTCON.GIE is executed before entering Standby mode or Halt mode, the next instruction will be executed after wake-up.

The user should add a NOP instruction immediately after the SLEEP instruction.

```
; Into Sleep Mode Method 1
MOVIA     C_Halt_Mode | C_FHOSC_Sel
MOVAR     OSCCR              ; Set OSCCR Register to Enters Halt Mode
NOP                          ; NOP Instruction After the Sleep Mode
; Into Sleep Mode Method 2
SLEEP                        ; Execute instruction to Enters Halt Mode
NOP                          ; NOP Instruction After the Sleep Mode
```

16.7 Summary of Operating Mode

The summary of four operating modes is described in the following table.

Mode	Normal	Slow	Standby	Halt
F_{HOSC}	Enabled	STPHOSC	STPHOSC	Disabled
F_{LOSC}	Enabled	Enabled	Enabled	Disabled
Instruction Execution	Executing	Executing	Stop	Stop
Timer0 / 1 / 4 / 5	TxEN	TxEN	TxEN	Disabled
WDT	Option and WDTEN	Option and WDTEN	Option and WDTEN	Option and WDTEN
Other Modules	Module enable bit	Module enable bit	Module enable bit	All disabled
Wake-up Source	-	-	- Timer0/1/4/5 overflow - WDT timeout - PA/PB/PC input change - INT0/1/2 - LVD interrupt - Comparator interrupt - ADC end-of-convert	- WDT timeout - PA/PB/PC input change - INT0/1/2

Table 16-1 Summary of Operating Modes

17. Reset Process

17.1 Overview

AT8F2481 Can be reset in the following 4 ways:

- Power-On Reset (POR) is occurred when VDD rising is detected.
- Low-Voltage Reset (LVR) is occurred when operating VDD is below pre-defined voltage.
- Pin RSTb is low state.
- WDT timeout reset.

Moreover, value of all registers will be initialized to their initial value or unchanged if its initial value is unknown. The status bits /TO and /PD could be initialized according to which event causes reset. The /TO and /PD value and its associated event is summarized in the table below.

Event	/TO	/PD
POR, LVR	1	1
RSTb reset from non-Halt mode	unchanged	unchanged
RSTb reset from Halt mode	1	1
WDT reset from non-Halt mode	0	1
WDT reset from Halt mode	0	0
SLEEP executed	1	0
CLRWDT executed	1	1

Table 17-1 Summary of /TO & /PD Value and its Associated Event

After Reset Event is released, AT8F2481 will start Reset Process. It will wait certain amount of period for oscillation stable no matter what kind of oscillator is adopted. This period is called power-up reset time and is determined by three-bit configuration words which can be 600us, 4.5ms, 18ms, 72ms or 288ms. After power-up reset time, AT8F2481 will wait for further oscillator start-up time (OST) before it starts to execute program. OST=1 clock cycle of F_{OSC} if the previous power-up time is 600us, OST=16 clock cycles of F_{OSC} if the previous power-up time is 4.5ms, 18ms, 72ms or 288ms.

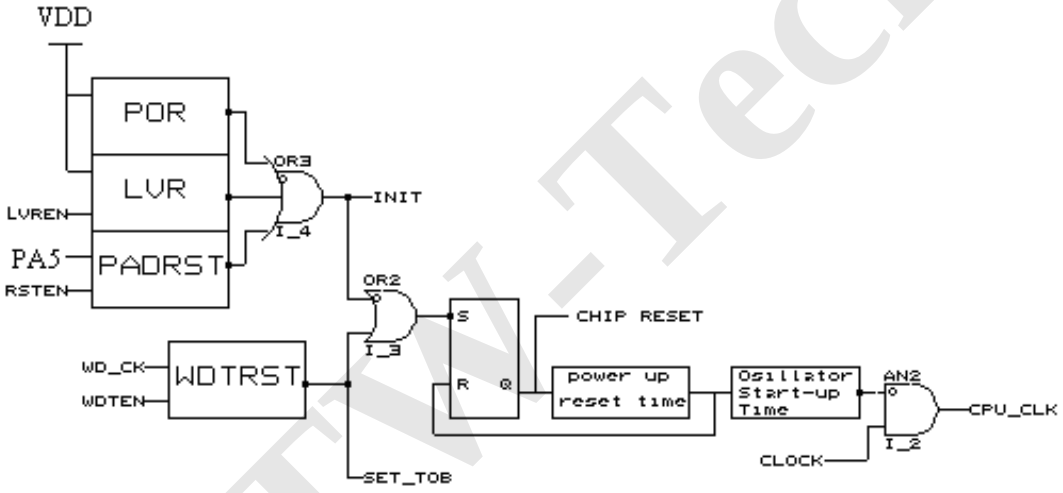


Figure 43 Block diagram of On-Chip Reset Circuit

PA5 can be used as external reset input determined by a configuration word. When an active-low signal is applied to PA5, it will cause AT8F2481 to enter reset process. For slow VDD power-up, it is recommended to use RSTb reset, as the following figure.

It is recommended the R value should be not greater than 40KΩ.

The R1 value=100Ω to 1KΩ will prevent high current, ESD or Electrical overstress flowing into reset pin.

The diode helps discharge quickly when power down.

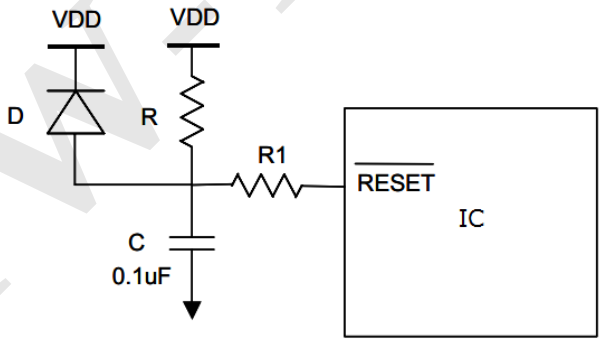


Figure 44 Block Diagram of Reset Application

18. SPI MODE

18.1 Overview

Features of the SPI include:

- Full-duplex operation
- Four programmable master mode frequencies
- Serial clock with programmable polarity and phase
- End of transmission interrupt flag
- Write collision error flag
- Bus contention error flag

AT8F2481 will enter SPI MODE by writing SIMCR[7:6]=10. It includes four line SPI interface if SIMCR[4] = 1,and It includes three line SPI interface if SIMCR[4]=0.

The communication is full duplex and operates as a slave/master type, where the device can be either master or slave. The device is master If SIMCR[5]=1, it is slave if SIMCR[5]=0

The SPI interface is a full duplex synchronous serial data link. It is a four-line interface with pin names SCK, MISO, MOSI and SSB. When choose three-line interface, SSB is omitted, no matter master or slave are always enable.

Only a master SPI can initiate transmissions. Software begins the transmission from a master SPI by writing to the SPI data register (SIMDR). The SIMDR does not buffer data being transmitted from the SPI. Data written to the SIMDR goes directly into the shift register and begins the transmission immediately under the control of the serial clock. The transmission ends after eight cycles of the serial clock when the SPI flag (SPIF) becomes set. At the same time that SPIF becomes set, the data shifted into the master SPI from the receiving device transfers to the SIMDR. The SIMDR buffers data being received by the SPI. Before the master SPI sends the next byte, software must clear the SPIF bit by reading the SPCR and then read or write the SIMDR.

In a slave SPI, data enters the shift register under the control of the serial clock from the master SPI. After a byte enters the shift register of a slave SPI, it transfers to the SIMDR. To prevent an overrun condition, slave software must then read the byte in the SIMDR before another byte enters the shift register and is ready to transfer to the SIMDR.

Read SPCR, and then read or write SIMDR will clear SPIF and WCOL.

Read SPCR, and then write SPCR will clear MODF

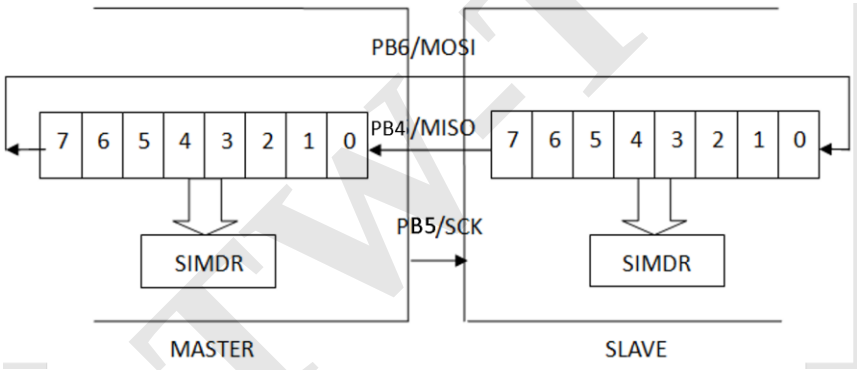


Figure 45 Single Master / Slave

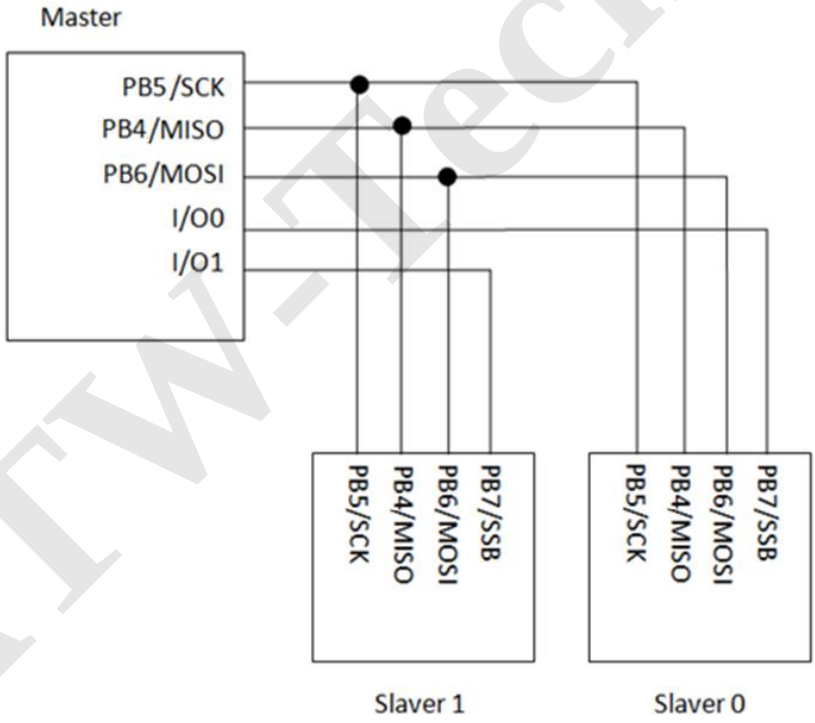


Figure 46 Single Master and Multi Slaves

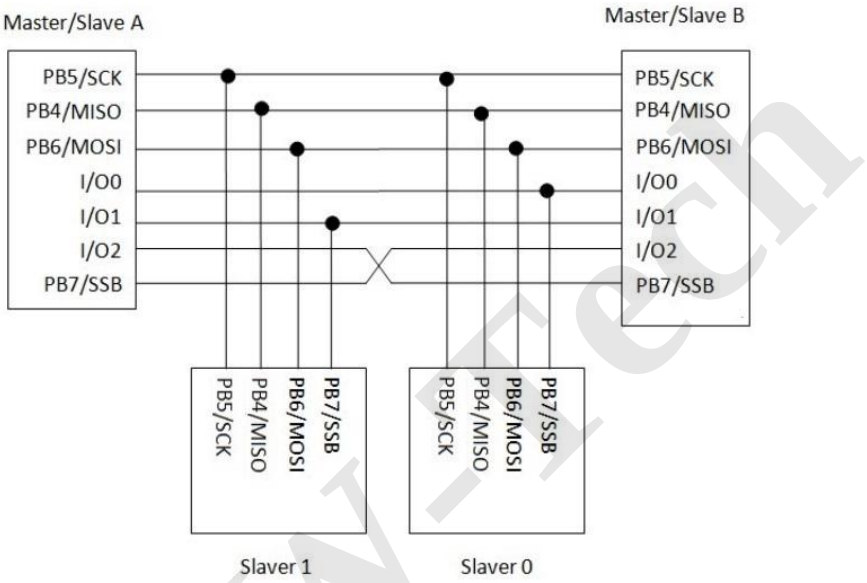


Figure 47 Multi Masters and Multi Slaves

18.2 Serial Clock Polarity and Phase

There are four kind of type to accommodate the different serial communication requirements of peripheral devices, depending upon the configurations of the CPOL (SPCR[3])bit and CKEG(SPCR[2]) bit.

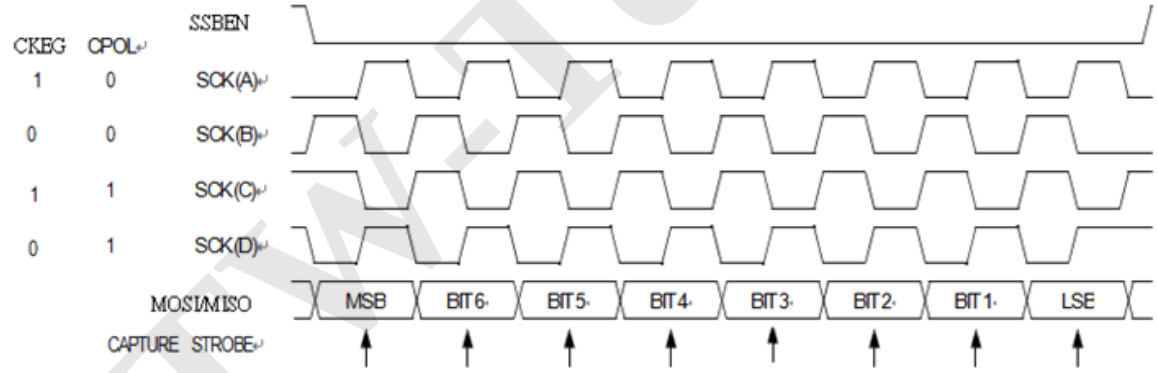


Figure 48 Shows How the CPOL and CKEG bits Affect the Clock/Data Timing

There are four kind of different SPI master clock rate, configuration by SPCR[1:0] (SPR[1:0]), SPR[1:0]=00/01/10/11 will choose system clock/2, system clock /4, system clock /16, system clock/32.

User must set up SPCR[3:0] firstly, and then enable SPI module by set SIMCR[7:6]=10, otherwise hardware errors will happen. The following picture is SPI flow chart

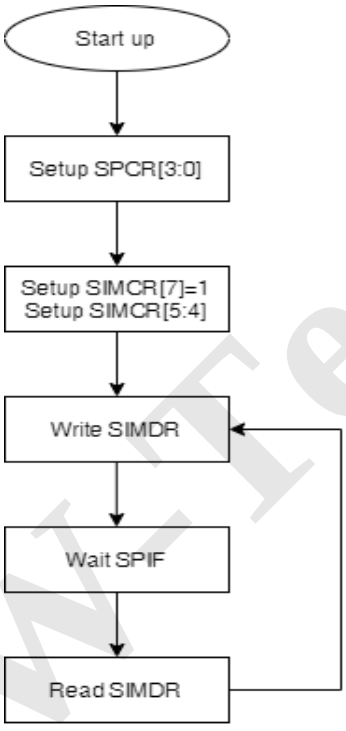


Figure 49 Flowchart of SPI Interface

18.3 SPI error Conditions

These conditions produce SPI system errors:

(1) MODF (Mode fault error) is happened if PB7/SSB is at logic 0 when in master mode.

(2) Writing to the SIMDR during a transmission causes a write-collision error and sets the WCOL bit in the SPCR. The error does not affect the transmission of the previously written byte, but the byte that caused the error is lost.

(3) Failing to read the SIMDR before the next incoming byte sets the SPIF bit

18.4 SPI Control Register

18.4.1 SIMDR (Serial Interface Mode Data Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SIMDR	0x112	SIMD7	SIMD6	SIMD5	SIMD4	SIMD3	SIMD2	SIMD1	SIMD0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		x	x	x	x	x	x	x	x

Bit 7:0 **SIMDR[7:0]:** 8bit Serial Interface Mode Data

If I²C mode enable, they are I²C mode data register

If SPI mode enable, they are SPI mode data register.

18.4.2 SIMCR (Serial Interface Mode Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SIMCR	0x113	SPE	MEN	MSTA	SSB_PADEN	RX_PADEN	TX_PADEN	RCLK_PADEN	UREN
R/W Property		R/W		R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7 **SPE :** SPI mode enable bit

1 = SPI mode enable

0 = SPI mode disable

Bit 6 **MEN :** I²C mode enable bit

1 = I²C mode enable

0 = I²C mode disable

Bit 5 **MSTA:** Selection of master mode / slave mode. (include I²C mode and spi mode)

1 = Select master mode

0 = Select slave mode

Bit 4 **SSB_PADEN :** Decide PB7 is pin share as SSB or not, this functionality is used at SPI mode.

1 = PB7 is used as SSB

0 = PB7 is not used as SSB (At that time, no matter master mode or slave, will always enable.)

Bit 3 **RX_PADEN:** UART interface RXPAD

1 = PB7 or PB0 is used as UART RX signal input

0 = PB7 or PB0 is used as general GPIO

Bit 2 **TX_PADEN:** UART interface TXPAD

1 = PB6 or PB1 is used as UART TX signal output

0 = PB6 or PB1 is used as general GPIO

Bit 1 **RCLK_PADEN:** UART interface RCLK_PADEN

1 = PB4 is used as UR RX CKT clock input, frequency =baud rate*/16

0 = PB4 is used as general GPIO

Bit 0 **UREN:** UART interface enable bit

1 = UART interface enable

0 = UART interface disable

18.4.3 SPCR (SPI Control & Status Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR	0x114	SPIF	WCOL	-	MODF	CPOL	CKEG	SPR[1:0]	
R/W Property		R	R	-	R	R/W	R/W	R/W	R/W
Initial Value		0	0	x	0	0	0	0	0

Bit 7 **SPIF**: SPI flag

1 = Transmission complete

0 = Transmission not complete

Read SPCR, and then read or write SIMDR will clear SPIF and WCOL.

Bit 6 **WCOL**: Write collision bit

1 = Invalid write to SIMDR

0 = No invalid write to SIMDR

Read SPCR, and then read or write SIMDR will clear SPIF and WCOL.

Bit 5 **Unimplemented**

Bit 4 **MODF**: Mode fault bit

1 = SSB pulled low while MSTR bit set

0 = SSB not pulled low while MSTR bit set

Read SPCR, and then write SPCR will clear MODF.

Bit 3 **CPOL**: Clock polarity bit

1 = SCK pin at logic 1 between transmissions, idle high

0 = SCK pin at logic 0 between transmissions, idle low

Bit 2 **CKEG**: SPI SCK clock active edge type selection

When CPOL = 1

1 = SCK is high base level and data capture at SCK falling edge

0 = SCK is high base level and data capture at SCK rising edge

When CPOL = 0

1 = SCK is low base level and data capture at SCK rising edge

0 = SCK is low base level and data capture at SCK falling edge

Bit 1:0 **SPR[1:0]**: SPI clock rate select bits

SPR[1:0]	SPI Clock Rate
00	system oscillation (F_{OSC})/2
01	system oscillation (F_{OSC})/4
10	system oscillation (F_{OSC})/16
11	system oscillation (F_{OSC})/32

19. I²C MODE

19.1 Overview

AT8F2481 will enter I²C MODE by writing SIMCR[7:6]=01. the I²C mode standard is a two wire bus. This two-wire bus minimizes the interconnection between devices and eliminates the need for address decoders, having the advantage of less PCB routing and economic hardware structure. I²C mode is suitable for applications requiring communications in a short distance among a number of devices.

I²C mode have the following feature

- Multi-master operation
- 32 software programmable serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost driven interrupt
- Calling address identification interrupt
- Generate/detect the start, stop and acknowledge signals
- Repeated START signal generation
- Bus busy detection

The following figure is I²C mode function block.

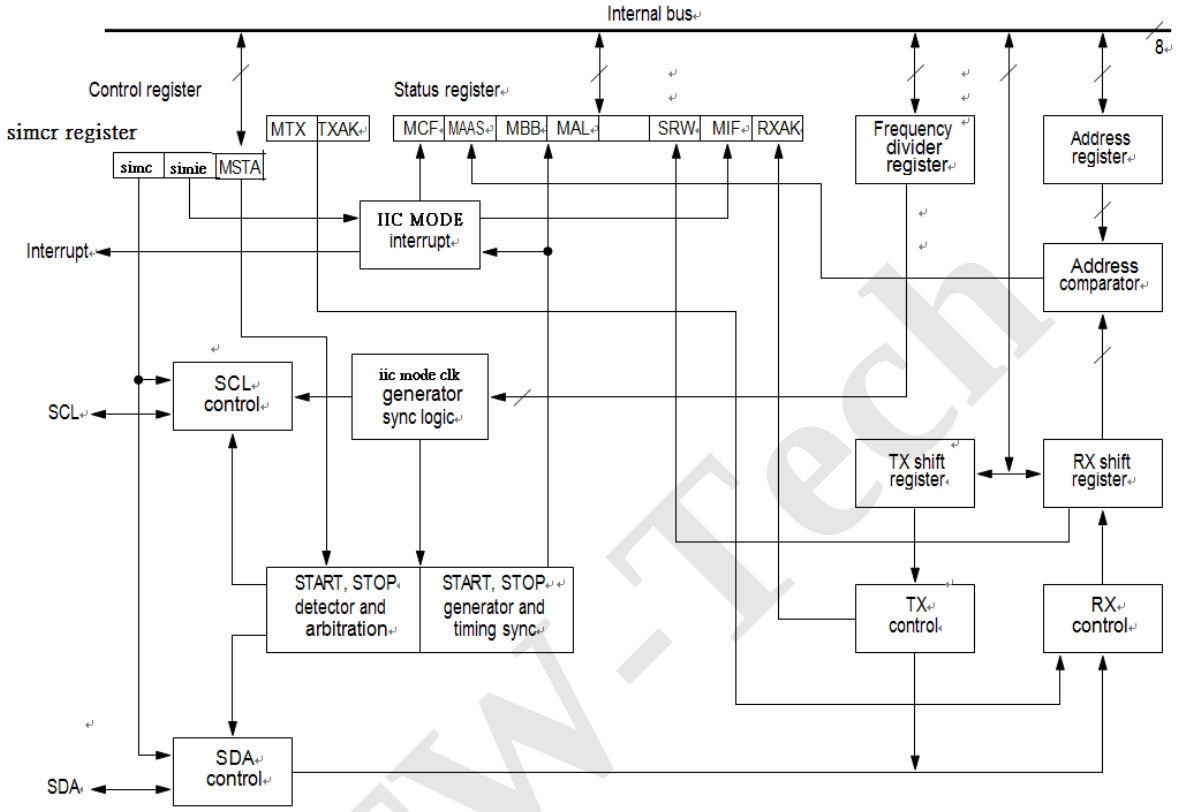


Figure 50 I²C Interface Block Diagram

The following figure is an example for I²C operation. One master device, and one slave device. SDA and SCK need pull high resistor.

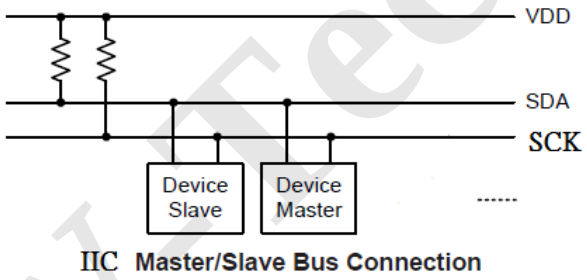


Figure 51 I²C Bus Connection

19.2 I²C Mode Protocol

A standard communication is composed of four parts, they are

(1) START

(2) slave address transfer

(3) data transfer

(4) STOP

The following diagram show SDA/SCK

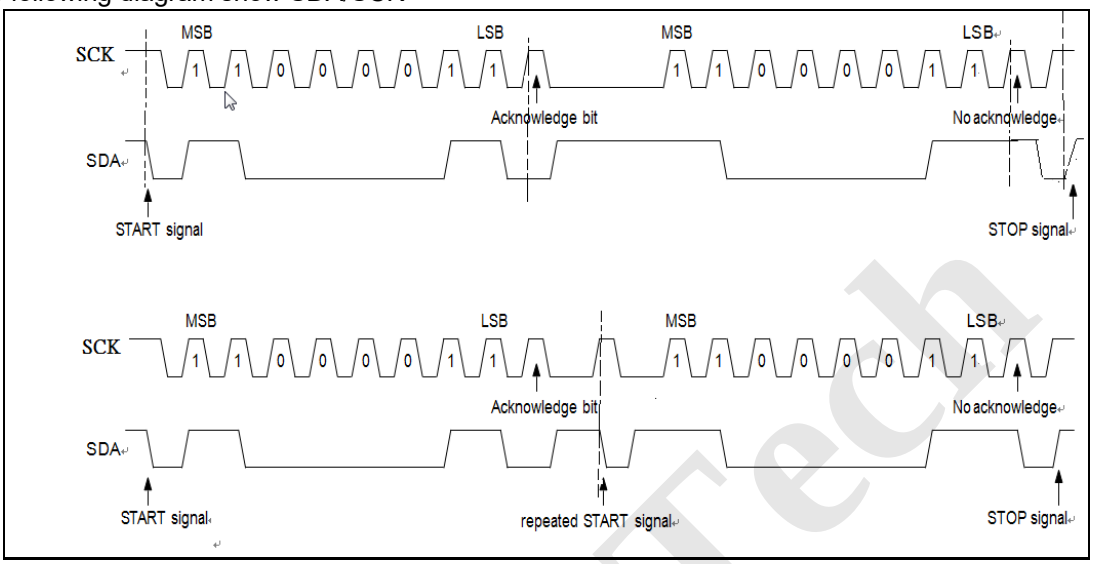


Figure 52 I²C Bus Transmission Signal Diagram

19.3 I²C Mode Operating

AT8F2481 will enter I²C MODE by writing SIMCR[7:6]=01The device will enter master mode if SIMCR[5](MSTA)=1, and The device will enter slave mode if SIMCR[5](MSTA)=0.

Upon reset, SIMCR[5] is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus, and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave.

MCR[4]=1, I²C MODE is set for transmit, otherwise MCR[4]=0, I²C MODE is set for receive mode .

Each time, we want to TX/RX data, firstly we go to clear MSR[1](MIF). And then Clear MSTA to 0 if MAL=1.

In Master/TX mode, after transmit 8 bit of data, will read MSR[0] to check receiving 9th clock sending acknowledge or not. If receiving acknowledge bit, then continues to write SIMDR, otherwise generating STOP signal to terminal communication.

IN master/RX mode, set TXAK=0 and receive data, it will send acknowledge signal at 9th clock bit. If last 2 byte of data have been read from SIMDR, and then set TXAK=1. If last bye has been read from SIMDR, then generate STOP signal.

IN Slave mode, check MSR[6] (MASS) if it is first byte. If MSR[6](MASS)=1, then continue to check if SRW is high or low. IF SRW is high, then set MCR[4](MTX)=1 to enter TX mode and write data to SIMDR ,else SRW is low, then set MCR[4](MTX)=0 to RX mode and dummy read SIMDR.

IN Slave/TX mode, after sending 8 bit of data, then read MSR[0](RXAK) to check if receiving 9th clock sending acknowledge or not. If receiving acknowledge bit, then continuing to write SIMDR, otherwise set RX mode and dummy read SIMDR.

The following diagram show the suggested flow.

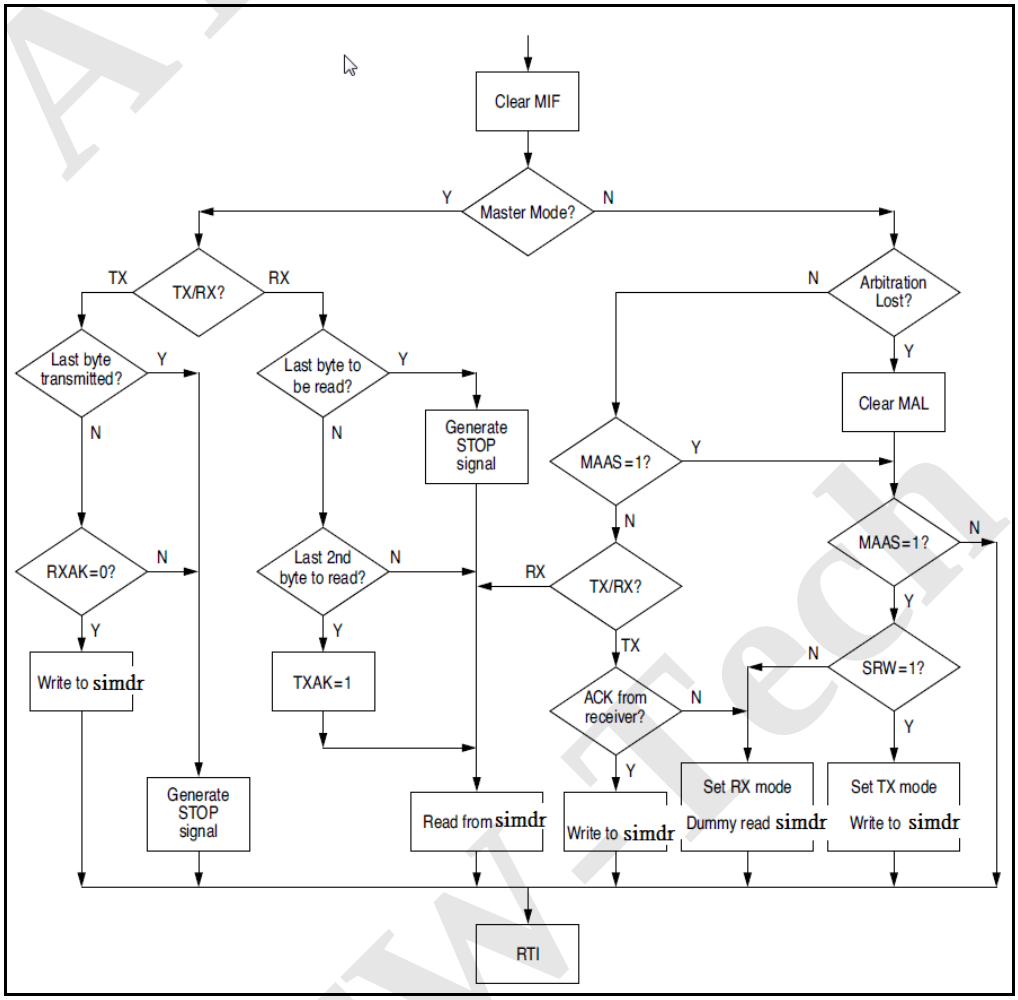


Figure 53 Flowchart of I²C Interface Routine

19.4 Arbitration Lost

This interface circuit is a true multi-master system which allows more than one master to be connected. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus

clock. The clock low period is equal to the longest clock low period among the masters; and the clock high period is the shortest among the masters. A data arbitration procedure determines the priority. A master will lose arbitration if it transmits a logic while the others transmit logic "0", the losing master will immediately switch over to slave receive mode and stops its data and clock outputs. The transition from master to slave mode will not generate a STOP condition. Meanwhile, a software bit will be set by hardware to indicate loss of arbitration.

19.5 I²C Control Register

19.5.1 MADR (I²C mode Address register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADR	0x115	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	-
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
Initial Value		0	0	0	0	0	0	0	x

Bit 7:1 **MAD[7:1]**: MAD[7:1] are the slave address bits of I²C mode

Bit 0 **Unimplemented**

19.5.2 MFDR (I²C mode frequency register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MFDR	0x116	-	-	-	FD4	FD3	FD2	FD1	FD0
R/W Property		-	-	-	R/W	R/W	R/W	R/W	R/W
Initial Value		x	x	x	0	0	0	0	0

Bit 7:5 **Unimplemented**

Bit 4:0 **FD[4:0]:** FD[4:0] are used for clock rate selection, the serilal bit clock frequency is equal to the system oscillation (F_{OSC}) divide by the divider shown in table as below.

Example: system oscillation (F_{OSC}) =1MHz, FD[4:0]=2,

I²C MODE CLK FREQ=1MHz/28=35.7KHz

FD[4:0]	DIVIDER	FD[4:0]	DIVIDER
00000	22	10000	352
00001	24	10001	384
00010	28	10010	448
00011	34	10011	544
00100	44	10100	704
00101	48	10101	768
00110	56	10110	896
00111	68	10111	1088
01000	88	11000	1408
01001	96	11001	1536
01010	112	11010	1792
01011	136	11011	2176
01100	176	11100	2816
01101	192	11101	3072
01110	224	11110	3584
01111	272	11111	4352

19.5.3 MCR (I²C mode control register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MCR	0x117	-	-	-	MTX	TXAK	-	-	-
R/W Property		-	-	-	R/W	R/W	-	-	-
Initial Value		x	x	x	0	0	x	x	x

Bit 7:5 **Unimplemented**

Bit 4 **MTX:** I²C mode Transmit/Receive Mode Select

1 = Transmit mode

0 = Receive mode

Bit 3 **TXAK:** I²C mode Acknowledge Enable

1 = Do not send acknowledge signal

0 = Send acknowledge at 9th clock bit

Bit 2:0 **Unimplemented**

19.5.4 MSR (I²C mode status register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MSR	0x118	MCF	MAAS	MBB	MAL	-	SRW	MIF	RXAK
R/W Property		R	R	R	R/W	-	R	R/W	R
Initial Value		1	0	0	0	x	0	0	1

Bit 7 **MCF:** Data Transfer Complete

1 = A byte has been completed

0 = A byte is being transfer

Bit 6 **MAAS:** Addressed as Slave

1 = Currently addressed as slave

0 = Not currently addressed

When MAAS is set, the MIF (I²C MODE interrupt) bit is also set.

If MAAS=1, Then CPU needs to check the SRW bit and set its MTX bit accordingly.

Bit 5 **MBB:** Bus Busy

1 = Bus busy

0 = Bus idle

Bit 4 **MAL:** Arbitration Lost

1 = Lost arbitration in master mode

0 = No arbitration lost

This arbitration lost flag is set when the I²C MODE master loses arbitration during a Master transmission mode. When MAL is set, the MIF bit is also set. This bit must be cleared by software

Bit 3 **Unimplemented**

Bit 2 **SRW :** Slave R/W select

1 = Read from slave, from calling master

0 = Write to slave from calling master

When MAAS is set, the R/W command bit of the calling address sent from the master is latched into this SRW bit. By checking this bit, device can then select slave transmit/receive mode by configuring MTX bit of the I²C MODE Control register

Bit 1 **MIF:** I²C interrupt flag

1 = I²C interrupt has occurred

0 = I²C interrupt has not occurred

When this bit is set, an interrupt is generated to the CPU if SIMIE is set. This bit is set when one of the following events occurs:

(1) Completion of one byte of data transfer. It is set at the falling edge of the 9th clock – MCF set.

(2) A match of the calling address with its own specific address in slave mode - MAAS set.

(3) A loss of bus arbitration - MAL set.

This bit must be cleared by software in the interrupt routine.

Bit 0 **RXAK:** Receive Acknowledge

1 = No acknowledgment signal detected

0 = Acknowledge signal detected after 8 bits' data transmitted

Note: The MIF and MAL bits are software clearable, while the other bits are read only.

19.6 Programming Consideration

19.6.1 Initialization

Reset will put the I²C data bus control register to its default status. Before the interface can be used to transfer serial data, the following initialization procedure must be carried out.

(1)Update Frequency Divider register (MFDR) to select an SCL frequency.

(2)Update I²C bus Address Register (MADR) to define its own slave address.

(3)Set SIMC0 bit of Serial Interface Control Register (SIMCR) to enable the I²C interface system.

(4)Modify the bits of Serial Interface Control Register (SIMCR) to select Master/Slave mode, Transmit/Receive, interrupt enable or not.

19.6.2 Generation of a START Signal and the First Byte of Data Transfer

After completion of the initialization procedure, serial data can be transmitted by selecting the master transmit mode. If the device is connected to a multi-master bus system, the state of the I²C bus busy (MBB) must be tested to check if the serial bus is free. If the bus is free (MBB=0), the START condition and the first byte (the slave address) can be sent. An example program which generate the START signal and transmits the first data byte (slave address) is shown below:

```
L_MBB_Check:
    BTRSC   MSR, MBB            ; Check MBB bit
    LGOTO   L_MBB_Check         ; Wait MBB=0

    BSR     SIMCR, IIC_En       ; Enable IIC interface
    BSR     SIMCR, Master_En    ; Set master mode

    MOVIA   0xA0
    MOVAR   SIMDR               ; Call the slave with address 0xA0
```

19.6.3 Software Responses after Transmission or Reception of one Byte

Upon the completion of the transmission or reception of a data byte, the data transferring bit (MCF) will be set, indicating one byte communication has been finished. The I^2C bus interrupt bit (MIF) will also be set to generate an I^2C bus interrupt if the interrupt is enabled. Software must clear the MIF bit in the interrupt routine first. The MCF bit can be cleared by reading the Serial Interface Data Register (SIMDR) in receive mode or writing to the SIMDR in transmit mode. Software may serve the Serial Interface Data in the program by monitoring the MIF bit if the interrupt is disabled. The following is an example of a software response by a master in transmit mode in the interrupt routine.

```
ISR:
    BCR     MSR, MIF            ; Clear MIF flag
    BTRSS   SIMCR, MSTA         ; Check MSTA bit
    LGOTO   Slave_Service       ; Branch if slave mode

    BTRSS   MCR, MTX            ; Check MTX bit
    LGOTO   Receive_Mode        ; Branch if in receive mode

    BTRSS   MSR, RXAK           ; Check ACK from receive
    LGOTO   End                 ; If no ACK end of transmission

TRANSMIT:
    MOVIA   0x55                ; Get the next byte of data
    MOVAR   SIMDR               ; Transmit the data
```

19.6.4 Generation of the STOP Signal

A data transfer ends with a STOP signal generated by the master device. A master in transmit mode can simply generate a STOP signal after all the data have been transmitted. The following is an example showing how a STOP condition is generated by a master in transmit mode.

```
    MOVIA   IIC_En
    MOVAR   SIMCR               ; Master mode change to slave mode for generate stop signal

    BCR     MSR, MIF            ; Clear MIF flag
```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data. This can be achieved by setting the transmit acknowledge bit (TXAK) before reading the 2nd last byte of data. Before reading the last byte of data, a STOP signal must be generated first. This following is an example showing how a STOP signal is generated by a master in receive mode.

For example: expected to receive 8 bytes of data

```
L_Receive_Service:
    MOVIA   0x07                ; Move 2nd last receive count to ACC
    SUBAR   R_RXCNT             ; Check receive count
    BTRSS   STATUS, Z           ; If receive count = 0x07, Z flag = 1, branch to response NACK
    LGOTO   L_ACK_Loop
    MOVIA   NACK
    MOVAR   MCR

L_ACK_Loop:
    MOVIA   0x08                ; Move last receive count to ACC
    SUBAR   R_RXCNT             ; Check receive count
    BTRSS   STATUS, Z           ; If receive count = 0x08, Z flag = 1, branch to receive end
    LGOTO   L_Receive_Service
    MOVIA   IIC_En
    MOVAR   SIMCR               ; Master mode change to slave mode for generate stop signal
    MOVIA   IIC_TX
    MOVAR   MCR                 ; Select TX mode
```

19.6.5 Generation of a Repeated START Signal

At the end of data transfer, if the master still wants to communicate on the bus, it can generate another START signal followed by another slave address without first generating a STOP signal. A program example is as shown.

```
    MOVIA   IIC_En
    MOVAR   SIMCR               ; Enable IIC interface
    BSR     SIMCR, Master_En    ; Set master mode

    MOVIA   0xA1
    MOVAR   SIMDR               ; Call the slave with address 0xA1
```

20. UART

20.1 Overview

The programmable asynchronous communications interface (UART) mega function provides data formatting and control to a serial communication channel.

The mega function has select, read/write, interrupt and bus interface logic features that allow data transfers over an 8-bit bi-directional parallel data bus system. With proper formatting and error checking, the mega function can transmit and receive serial data, supporting asynchronous operation.

- Full double buffering
- Asynchronous operation
- Independently controlled Transmit, Line Status and Receive Interrupts
- Programmable data word length (5 - 8 bit), parity and stop bits
- Parity, overrun and framing error checking
- Programmable Baud Rate Generator allows division of any reference clock by 1 to (216-1) and generates an internal 16 x Clock.
- False start bit detection
- Automatic break generation and detection
- Internal diagnostic capabilities

The transmitter section is composed of a Transmit Holding Register (THR) and a Transmit Shift Register (TSR). Writing to THR will transfer the contents of the data bus (DIN 7-0) to the Transmit Holding Register every time that the THR or TSR is empty. This write operation should be done when Transmit Holding Register Empty (THRE) is set

This register contains the assembled received data. On the falling edge of the start bit, the receiver section starts its operations. The start bit is valid if the RXDATA is still low at the middle sample of Start bit, thus preventing the receiver from assembling a false data character

The Line Control Register is used to specify the data communication format. The break feature, parity, stop bits and word length can be changed by writing to the appropriate bits in LSR.

20.2 UART Control Register

20.2.1 DLL (Baud Rate Divisor Latch LSB Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DLL	0x119	DLL7	DLL6	DLL5	DLL4	DLL3	DLL2	DLL1	DLL0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7:0 **DLL[7:0]:** Baud rate divider LSB bit

20.2.2 DLH (Baud Rate Divisor Latch MSB Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DLH	0x11A	DLH7	DLH6	DLH5	DLH4	DLH3	DLH2	DLH1	DLH0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7:0 **DLH[7:0]:** Baud rate divider MSB bit

Baud rate=I_HRC / [16x(N)], N={DLH[7:0], DLL[7:0]}

20.2.3 LCR (Line Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCR	0x11B	LOOP	SBRK	PSTUCK	PEVEN	PREN	STPS	WL1	WL0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7 **LOOP:** Loop back test enable bit

1 = The output of the transmitter shift is loop back to the receiver shift register input

0 = Disable the loop back test

Bit 6 **SBRK:** Set break bit

1 = The serial output is forced to the spacing (logic 0) state and remains there regardless of other transmitter activity

Bit 5 **PSTUCK:** Stuck parity bit

1 = The parity bit is sent and then detected by the receiver as a logic 0

0 = The parity bit is logic 1

Bit 4 **PEVEN:** Even parity select bit

1 = An even number of bits is sent or checked

0 = An odd number of bits is sent or checked

Bit 3 **PREN:** Parity enable bit

1 = A parity bit is generated (transmit data) or check (receive data) between the last data word and stop bit of the serial data

0 = No parity is generated

Bit 2 **STPS:** Number of STOP bits select

STPS	Word Length	Number of Stop bits
0	X	1
1	5	1.5
1	6,7,8	2

Bit 1:0 **WL[1:0]:** Word length select bits

WL[1:0]	Word length bits
00	5 bits per character
01	6 bits per character
10	7 bits per character
11	8 bits per character

20.2.4 LSR (Line Status Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSR	0x11C	-	TSRE	THRE	BKINT	FERR	PERR	OVERR	READY
R/W Property		-	R	R	R	R	R	R	R
Initial Value		x	1	1	0	0	0	0	0

Bit 7 **Unimplemented**

Bit 6 **TSRE:** Transmitter shift register(TSR) empty:

This bit is set to logical 1 whenever the transmitter holding register and the transmitter shift

register are both empty. It is reset to logical 0 whenever THR or TSR contains a data character. ***Write THR/RBR will clear TSRE***.

Bit 5 **THRE:** Transmitter holding register (THR) empty flag

This bit indicates the controller is ready to accept a new character for transmission. It is set to logic 1 when a character is transferred from the transmitter holding register (THR) into the transmitter shift register. ***Write THR/RBR will clear THRE.***

Bit 4 **BKINT:** Break interrupt flag

This bit is set to logical 1 whenever the receiver data input is held in the spacing state (logical 0) for longer than a full word transmission time. ***Read LSR Clear BKINT.***

Bit 3 **FERR:** Frame error flag

This bit indicates the received character did not have a valid stop bit. It is set to logical 1 whenever the stop bit following the last data bit or parity bit is detected as a zero bit.

Read LSR Clear FERR.

Bit 2 **PERR:** Parity error flag

1 = Detect a parity error

0 = No parity error

Read LSR Clear PERR.

Bit 1 **OVERR:** Over run error flag

1 = Over run happen

0 = Not happen over run

Read LSR Clear OVERR.

Bit 0 **READY:** Data ready flag bit

It is set to logic 1 whenever a complete incoming character has been received and transferred into the receiver buffer register.

Read the THR/RBR data will clear this flag.

20.2.5 THR/RBR (Transmit holding register /Receive Buffer Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
THR/RBR	0x11D	URD7	URD6	URD5	URD4	URD3	URD2	URD1	URD0
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		x	x	x	x	x	x	x	x

Bit 7:0 **URD[7:0]:** f write data to the register, they are UART transmit data register

If read data from the register, they are UART receive data register.

21. LCD

21.1 Overview

AT8F2481 has a built-in LCD driver module. LCD driver module is used to drive ½ VDD bias to one of 8 COM PORTs when LCDEN is set to 1 and the corresponding PxCOM bit in SFR LDCCON1 is set to 1. These 8 COM PORTs and the corresponding PxCOM are listed in the following table:

COM PORT	SFR PxCOM
PB4	P0COM
PB5	P1COM
PB6	P2COM
PB7	P3COM
PC2	P4COM
PC3	P5COM
PC4	P6COM
PC5	P7COM

Table 21-1 LCD COM Port Select

The SEG PORT of AT8F2481 can be any IO pads who can export voltage level of VDD and VSS.

A complete LCD waveform cycle contains two frames, namely, Frame P and Frame N as the following figure shows 3 COMs LCD waveforms.

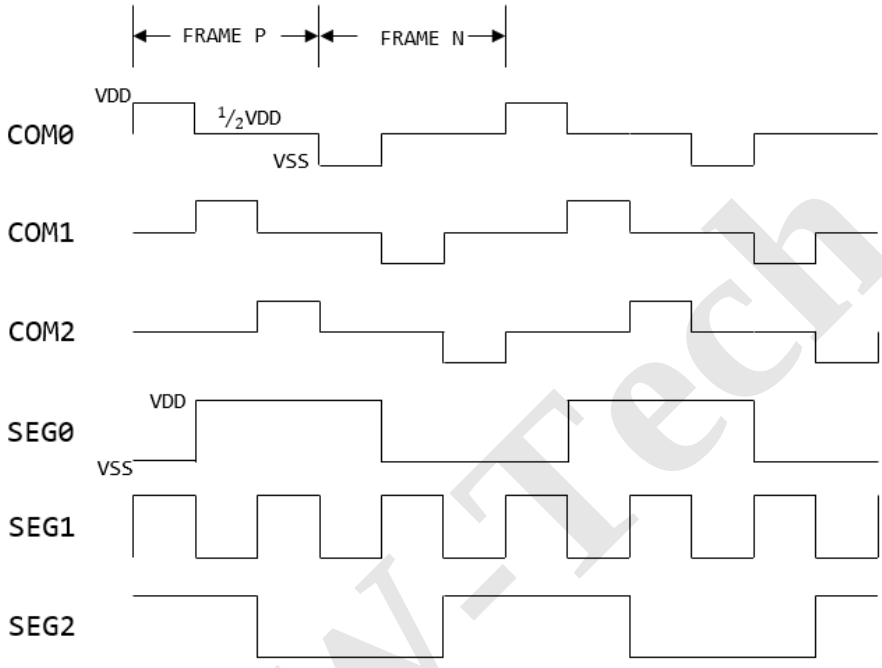


Figure 54 LCD Waveform

To Lit up the corresponding COM/SEG pair, set SEG PAD to VSS and COM PAD to VDD In Frame P, and likewise set SEG PAD to VDD and COM PAD to VSS in Frame N. To be specifically, refer to the following table for LCD pad setting:

PADS	Light		Dark	
	Frame P	Frame N	Frame P	Frame N
COM	PxCOM=0 mode=output output=VDD	PxCOM=0 mode=output output=VSS	PxCOM=0 mode=output output=VDD	PxCOM=0 mode=output output=VSS
SEG	mode=output output=VSS	mode=output output=VDD	mode=output output=VSS	mode=output output=VDD

Table 21-2 LCD PAD Setting

21.2 LCD Control Register

21.2.1 LCDCON0 (LCD Control Register 0)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON0	0x18C	LCDEN	-	-	-	-	-	LCDS1	LCDS0
R/W Property		R/W	-	-	-	-	-	R/W	R/W
Initial Value		0	x	x	x	x	x	0	0

Bit 7 **LCDEN:** LCD function enable

1 = LCD function is enabled

0 = LCD function is disabled

Bit 6 :2 **Unimplemented**

Bit 1 :0 **LCDS[1:0]:** ½ VDD current select

11 = Current=100 uA

10 = Current=50 uA

01 = Current=16 uA

00 = Current=8 uA

21.2.2 LCDCON1 (LCD Control Register 1)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON1	0x18D	P7COM	P6COM	P5COM	P4COM	P3COM	P2COM	P1COM	P0COM
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7 **P7COM:** ½ LCD bias selection

1 = PC5 output ½ VDD

0 = PC5 as normal GPIO

Bit 6 **P6COM:** ½ LCD bias selection

1 = PC4 output ½ VDD

0 = PC4 as normal GPIO

Bit 5 **P5COM:** ½ LCD bias selection

1 = PC3 output ½ VDD

0 = PC3 as normal GPIO

Bit 4 **P4COM:** ½ LCD bias selection

1 = PC2 output ½ VDD

0 = PC2 as normal GPIO

Bit 3 **P3COM:** ½ LCD bias selection

1 = PB7 output ½ VDD

0 = PB7 as normal GPIO

Bit 2 **P2COM:** ½ LCD bias selection

1 = PB6 output ½ VDD

0 = PB6 as normal GPIO

Bit 1 **P1COM:** ½ LCD bias selection

1 = PB5 output ½ VDD

0 = PB5 as normal GPIO

Bit 0 **P0COM:** ½ LCD bias selection

1 = PB5 output ½ VDD

0 = PB5 as normal GPIO

Note when PxCOM=1, the corresponding PAD output buffer is disabled automatically.

22. Operational Amplifier(OPA0 and OPA1)

22.1 Overview

NF8F2481 has built in two operation amplifier OPA0 and OPA1. The function and performances of two OP are the same. OPA0/OPA1 has the following feature.

- Internal integrated zero adjustment circuit
- The positive and negative terminals can be connected to the I/O port.
- The output terminal can be connected to the I/O port or the internal ADC detection channel
- Can be used as a comparator.

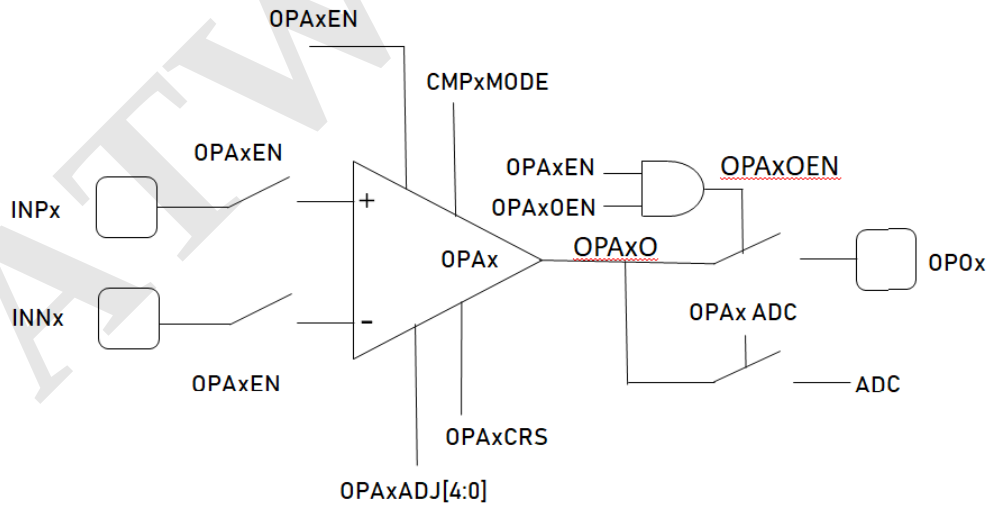


Figure 55 OPAx BLOCK

After the OPAxEN is enabled, the positive and negative terminals are automatically connected to the I/O port. Set the OPAxOEN to 1, the OPA output can be connected to the I/O port. The OPAx output can be connected to ADC channel by setting the OPAx_ADC to 1.

The OPAx have normal mode and adjustment mode. When the OPAxCOFM bit is set to 1, the OPAx enter the adjustment mode. In the adjustment mode, the positive and negative terminals of the OPAx are internally short-circuited and connected to the positive or negative side of the OPAx selected by OPAxCRS bit.

By setting the CMPxMODE to 1, The OPAx can be used as comparator.

Note:

1. At the same time, only one OPAxO output can be connected to ADC channel, and two OPAxO output cannot be connected to ADC channel at the same time.

2. When OPAxEN=1 and OPAxOEN=1, The Hardware related I/O ports used by OPAx will be set to input state.

22.2 OPA Control Register

22.2.1 OP (OP Amplifier Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPACON	0x188	OPA1EN	OPA1OEN	CMP1MODE	OPA1_ADC	OPA0EN	OPA0OEN	CMP0MODE	OPA0_ADC
R/W Property		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	0	0	0	0	0

Bit 7 **OPA1EN:** OPA1 enable bit

1 = OPA1 enable

0 = OPA1 disable

Bit 6 **OPA1OEN:** OPA1 output enable

1 = OPA1 output is connected to the I/O port (OPA0O pin)

0 = OPA1 output is not connected to the I/O port

Bit 5 **CMP1MODE:** Comparator mode selection

1 = Comparator mode

0 = OP AMP mode

Bit 4 **OPA1_ADC:** ADC channel enable

1 = OPA1 output is connected to OPA1_ADC channel

0 = OPA1 output is not connected to OPA1_ADC channel

Bit 3 **OPA0EN:** OPA0 enable bit

1 = OPA0 enable

0 = OPA0 disable

Bit 2 **OPA0OEN:** OPA0 output enable

1 = OPA0 output is connected to the I/O port (OPA0O pin)

0 = OPA0 output is not connected to the I/O port

Bit 1 **CMP0MODE:** Comparator mode selection

1 = Comparator mode

0 = OP AMP mode

Bit 0 **OPA0_ADC:** ADC channel enable

1 = OPA0 output is connected to OPA0_ADC channel

0 = OPA0 output is not connected to OPA0_ADC channel

22.2.2 OPA0ADJ (OPA0 Adjustment Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPA0ADJ	0x19E	OPA0 DOUT	OPA0 COFM	OPA0 CRS	OPA0 ADJ4	OPA0 ADJ3	OPA0 ADJ2	OPA0 ADJ1	OPA0 ADJ0
R/W Property		R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	Trim Code				

Bit 7 **OPA0DOUT:** OPA0 output result

1 = The op amp output is high, and the positive terminal is higher than the negative terminal voltage

0 = The op amp output is low, and the positive terminal is lower than the negative terminal voltage

Bit 6 **OPA0COFM:** Comparator mode selection

1 = OPA0 working in adjustment mode

0 = OPA0 working in normal mode; CMP Mode used

Bit 5 **OPA0CRS:** OPA0 adjustment mode input selection bit

1 = OPA0 adjustment mode positive input

0 = OPA0 adjustment mode negative input

Bit 4:0 **OPA0ADJ[4:0]:** OPA0 offset voltage adjustment bit (Trim Code)

22.2.3 OPA1ADJ (OPA1 Adjustment Control Register)

Name	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPA1ADJ	0x19F	OPA1DOUT	OPA1COFM	OPA1CRS	OPA1ADJ4	OPA1ADJ3	OPA1ADJ2	OPA1ADJ1	OPA1ADJ0
R/W Property		R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value		0	0	0	Trim Code				

Bit 7 **OPA1DOUT:** OPA1 output result

1 = The op amp output is high, and the positive terminal is higher than the negative terminal voltage

0 = The op amp output is low, and the positive terminal is lower than the negative terminal voltage

Bit 6 **OPA1COFM:** Comparator mode selection

1 = OPA1 working in adjustment mode

0 = OPA1 working in normal mode; CMP Mode used

Bit 5 **OPA1CRS:** OPA1 adjustment mode input selection bit

1 = OPA1 adjustment mode positive input

0 = OPA1 adjustment mode negative input

Bit 4:0 **OPA1ADJ[4:0]:** OPA1 offset voltage adjustment bit (Trim Code)

23. Instruction Set

AT8F2481 provides 40 powerful instructions for all kinds of applications.

Inst.	OP 1	OP 2	Operation	Cyc.	Flag
Arithmetic Instructions					
ANDAR	R	d	dest = ACC & R	1	Z
IORAR	R	d	dest = ACC | R	1	Z
XORAR	R	d	dest = ACC ⊕ R	1	Z
ANDIA	i		ACC = ACC & i	1	Z
IORIA	i		ACC = ACC | i	1	Z
XORIA	i		ACC = ACC ⊕ i	1	Z
RRR	R	d	Rotate right R	1	C
RLR	R	d	Rotate left R	1	C
BSR	R	bit	Set bit in R	1	-
BCR	R	bit	Clear bit in R	1	-
INCR	R	d	Increase R	1	Z
DECR	R	d	Decrease R	1	Z
COMR	R	d	dest = ~R	1	Z
Conditional Instructions					
BTRSC	R	bit	Test bit in R, skip if clear	1 or 2	-
BTRSS	R	bit	Test bit in R, skip if set	1 or 2	-
INCRSZ	R	d	Increase R, skip if 0	1 or 2	-
DECRSZ	R	d	Decrease R, skip if 0	1 or 2	-
Data Transfer Instructions					
MOVAR	R		Move ACC to R	1	-
MOVR	R	d	Move R	1	Z
MOVIA	i		Move immediate to ACC	1	-
SWAPR	R	d	Swap halves R	1	-
TABLEA			Read ROM	2	-

Inst.	OP 1	OP 2	Operation	Cyc.	Flag
Arithmetic Instructions					
ADDAR	R	d	dest = R + ACC	1	Z, DC, C
SUBAR	R	d	dest = R + (~ACC)	1	Z, DC, C
ADCAR	R	d	dest = R + ACC + C	1	Z, DC, C
SBCAR	R	d	dest = R + (~ACC) + C	1	Z, DC, C
ADDIA	i		ACC = i + ACC	1	Z, DC, C
SUBIA	i		ACC = i + (~ACC)	1	Z, DC, C
ADCIA	i		ACC = i + ACC + C	1	Z, DC, C
SBCIA	i		ACC = i + (~ACC) + C	1	Z, DC, C
CLRA			Clear ACC	1	Z
CLRR			Clear R	1	Z
Other Instructions					
NOP			No operation	1	-
SLEEP			Go into Halt mode	1	/TO, /PD
CLRWDT			Clear Watch-Dog Timer	1	/TO, /PD
RET			Return from subroutine	2	-
RETIE			Return from interrupt and enable interrupt	2	-
RETIA	i		Return, place immediate in ACC	2	-
LCALL	adr		Call subroutine(4K)	2	-
LGOTO	adr		unconditional branch(4K)	2	-

Table 23-1 Instruction Set

ACC: Accumulator.
adr: immediate address.
bit: bit address within an 8-bit register R.
C: Carry/Borrow bit
C=1, carry is occurred for addition instruction or borrow is **NOT** occurred for subtraction instruction.
C=0, carry is not occurred for addition instruction or borrow **IS** occurred for subtraction instruction.

d: Destination

If d is “0”, the result is stored in the ACC.

f d is “1”, the result is stored back in register R.

DC: Digital carry flag.
dest: Destination.

i: 8-bit immediate data.

PC: Program Counter.
PCHBUF: High Byte Buffer of Program Counter.

/PD: Power down flag bit

/PD=1, after power-up or after instruction CLRWDT is executed.

/PD=0, after instruction SLEEP is executed.

Prescaler: Prescaler0 dividing rate.

R: SFR, R is 0x0 ~0x7F.

T0MD: T0MD register.

TBHP: The high-Byte at target address in ROM.

TBHD: Store the high-Byte data at target address in ROM.

/TO: Time overflow flag bit
/TO=1, after power-up or after instruction CLRWDT or SLEEP is executed.
/TO=0, WDT timeout is occurred.

WDT: Watchdog Timer Counter.

Z: Zero flag

ADCAR	Add ACC and R with Carry
Syntax:	ADCAR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	R + ACC + C dest
Status affected:	Z, DC, C
Description:	Add the contents of ACC and register R with Carry. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle	1
Example:	ADCAR R, d before executing instruction: ACC=0x12, R=0x34, C=1, d=1, after executing instruction: R=0x47, ACC=0x12, C=0.

ADDAR	Add ACC and R
Syntax:	ADDAR R, d
Operand:	0≦R ≦511 d = 0, 1.
Operation:	ACC + R→dest
Status affected:	Z, DC, C
Description:	Add the contents of ACC and R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	ADDAR R, d before executing instruction: ACC=0x12, R=0x34, C=1, d=1, after executing instruction: R=0x46, ACC=0x12, C=0.

ADCIA	Add ACC and Immediate with Carry
Syntax:	ADCIA i
Operand:	0 ≦ i < 255
Operation:	ACC + i + C ACC
Status affected:	Z, DC, C
Description:	Add the contents of ACC and the 8-bit immediate data i with Carry. The result is placed in ACC.
Cycle:	1
Example:	ADCIA i before executing instruction: ACC=0x12, i=0x34, C=1, after executing instruction: ACC=0x47, C=0.

ADDIA	Add ACC and Immediate
Syntax:	ADDIA i
Operand:	0 ≦ i < 255
Operation:	ACC + i ACC
Status affected:	Z, DC, C
Description:	Add the contents of ACC with the 8-bit immediate data i. The result is placed in ACC.
Cycle:	1
Example:	ADDIA i before executing instruction: ACC=0x12, i=0x34, C=1, after executing instruction: ACC=0x46, C=0.

ANDAR	AND ACC and R
Syntax:	ANDAR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	ACC & R dest
Status affected:	Z
Description:	The content of ACC is AND'ed with R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	ANDAR R, d before executing instruction: ACC=0x5A, R=0xAF, d=1. after executing instruction: R=0x0A, ACC=0x5A, Z=0.

ANDIA	AND Immediate with ACC
Syntax:	ANDIA i
Operand:	0 ≦ i < 255
Operation:	ACC & i→ACC
Status affected:	Z
Description:	The content of ACC register is AND' ed with the 8-bit immediate data i. The result is placed in ACC.
Cycle:	1
Example:	ANDIA i before executing instruction: ACC=0x5A, i=0xAF, after executing instruction: ACC=0x0A, Z=0.

BCR	Clear Bit in R
Syntax:	BCR R, bit
Operand:	0 ≦ R ≦ 511 0 ≦ bit ≦ 7
Operation:	0 R[bit]
Status affected:	--
Description:	Clear the bitth position in R.
Cycle:	1
Example:	BCR R, B2 before executing instruction: R=0x5A, B2=0x3, after executing instruction: R=0x52.

BSR	Set Bit in R
Syntax:	BSR R, bit
Operand:	0≦ R ≦ 511 0 ≦ bit ≦ 7
Operation:	1→R[bit]
Status affected:	--
Description:	Set the bitth position in R.
Cycle:	1
Example:	BSR R, B2 before executing instruction: R=0x5A, B2=0x2, after executing instruction: R=0x5E.

BTRSC	Test Bit in R and Skip if Clear
Syntax:	BTRSC R, bit
Operand:	0 ≦ R ≦ 511 0 ≦ bit ≦ 7
Operation:	Skip next instruction, if R[bit] = 0.
Status affected:	--
Description:	If R[bit] = 0, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	BTRSC R, B2 Instruction1 Instruction2 before executing instruction: R=0x5A, B2=0x2, after executing instruction: because R[B2]=0, instruction1 will not be executed, the program will start execute instruction from instruction2.

BTRSS	Test Bit in R and Skip if Set
Syntax:	BTRSS R, bit
Operand:	0 ≦ R ≦ 511 0 ≦ bit ≦ 7
Operation:	Skip next instruction, if R[bit] = 1.
Status affected:	--
Description:	If R[bit] = 1, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	BTRSS R, B2 Instruction2 Instruction3 before executing instruction: R=0x5A, B2=0x3, after executing instruction: because R[B2]=1, instruction2 will not be executed, the program will start execute instruction from instruction3.

CLRA	Clear ACC
Syntax:	CLRA
Operand:	--
Operation:	00h→ACC 1→Z
Status affected:	Z
Description:	ACC is clear and Z is set to 1.
Cycle:	1
Example:	CLRA before executing instruction: ACC=0x55, Z=0. after executing instruction: ACC=0x00, Z=1.

CLRR	Clear R
Syntax:	CLRR R
Operand:	0 ≦ R ≦ 127
Operation:	00h→R 1→Z
Status affected:	Z
Description:	The content of R is clear and Z is set to 1.
Cycle:	1
Example:	CLRR R before executing instruction: R=0x55, Z=0. after executing instruction: R=0x00, Z=1.

CLRWDT	Clear Watch-Dog Timer
Syntax:	CLRWDT
Operand:	--
Operation:	00h→WDT, 00h→WDT prescaler 1→/TO 1→/PD
Status affected:	/TO, /PD
Description:	Executing CLRWDT will reset WDT, Prescaler0 if it is assigned to WDT. Moreover, status bits /TO and /PD will be set to 1.
Cycle:	1
Example:	CLRWDT before executing instruction: /TO=0 after executing instruction: /TO=1

COMR	Complement R
Syntax:	COMR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	~R→dest
Status affected:	Z
Description:	The content of R is complemented. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	COMR, d before executing instruction: R=0xA6, d=1, Z=0. after executing instruction: R=0x59, Z=0.

DECR	Decrease R
Syntax:	DECR R, d
Operand:	0 ≦ R ≦ 127 d = 0, 1.
Operation:	R - 1→dest
Status affected:	Z
Description:	Decrease R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	DECR R, d before executing instruction: R=0x01, d=1, Z=0. after executing instruction: R=0x00, Z=1.

DECRSZ	Decrease R, Skip if 0
Syntax:	DECRSZ R, d
Operand:	0 ≦ R ≦ 127 d = 0, 1.
Operation:	R - 1→dest, Skip if result = 0
Status affected:	--
Description:	Decrease R first. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R. If result is 0, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	DECRSZ R, d instruction2 instruction3 before executing instruction: R=0x1, d=1, Z=0. after executing instruction: R=0x0, Z=1, and instruction will skip instruction2 execution because the operation result is zero.

INCR	Increase R
Syntax:	INCR R, d
Operand:	0 ≦ R ≦ 127 d = 0, 1.
Operation:	R + 1→dest.
Status affected:	Z
Description:	Increase R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	INCR R, d before executing instruction: R=0xFF, d=1, Z=0. after executing instruction: R=0x00, Z=1.

IORAR	OR ACC with R
Syntax:	IORAR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	ACC | R→dest
Status affected:	Z
Description:	OR ACC with R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	IORAR R, d before executing instruction: R=0x50, ACC=0xAA, d=1, Z=0. after executing instruction: R=0xFA, ACC=0xAA, Z=0.

INCRSZ	Increase R, Skip if 0
Syntax:	INCRSZ R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	R + 1→dest, Skip if result = 0
Status affected:	--
Description:	Increase R first. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R. If result is 0, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	INCRSZ R, d instruction2, instruction3. before executing instruction: R=0xFF, d=1, Z=0. after executing instruction: R=0x00, Z=1. And the program will skip instruction2 execution because the operation result is zero.

IORIA	OR Immediate with ACC
Syntax:	IORIA i
Operand:	0 ≦ i < 255
Operation:	ACC | i→ACC
Status affected:	Z
Description:	OR ACC with 8-bit immediate data i. The result is stored in ACC.
Cycle:	1
Example:	IORIA i before executing instruction: i=0x50, ACC=0xAA, Z=0. after executing instruction: ACC=0xFA, Z=0.

LCALL	Call Subroutine
Syntax:	LCALL adr
Operand:	0 ≦ adr ≦ 4095
Operation:	PC + 1→Top of Stack, adr→PC[11:0]
Status affected:	--
Description:	The return address (PC + 1) is pushed onto top of Stack. The 12-bit immediate address (adr) is loaded into PC[11:0].
Cycle:	2
Example:	LCALL SUB before executing instruction: PC=A0. Stack level=1 after executing instruction: PC=address of SUB, Stack[1]= A0+1, Stack pointer =2.

LGOTO	Unconditional Branch
Syntax:	LGOTO adr
Operand:	0 ≦ adr ≦4095
Operation:	adr→PC[11:0].
Status affected:	--
Description:	LGOTO is an unconditional branch instruction. The 12-bit immediate address (adr) is loaded into PC[11:0].
Cycle:	2
Example:	LGOTO Level before executing instruction: PC=A0. after executing instruction: PC=address of Level.

MOVAR	Move ACC to R
Syntax:	MOVAR R
Operand:	0 ≦ R ≦ 511
Operation:	ACC→R
Status affected:	--
Description:	Move content of ACC to R.
Cycle:	1
Example:	MOVAR R before executing instruction: R=0x55, ACC=0xAA. after executing instruction: R=0xAA, ACC=0xAA.

MOVIA	Move Immediate to ACC
Syntax:	MOVIA i
Operand:	0 ≦ i < 255
Operation:	i→ACC
Status affected:	--
Description:	The content of ACC is loaded with 8-bit immediate data i.
Cycle:	1
Example:	MOVIA i before executing instruction: i=0x55, ACC=0xAA. after executing instruction: ACC=0x55.

MOVR	Move R to ACC or R
Syntax:	MOVR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	R→dest
Status affected:	Z
Description:	The content of R is move to destination. If d is 0, destination is ACC. If d is 1, destination is R and it can be used to check whether R is zero according to status flag Z after execution.
Cycle:	1
Example:	MOVR R, d before executing instruction: R=0x0, ACC=0xAA, Z=0, d=0. after executing instruction: R=0x0, ACC=0x00, Z=1.

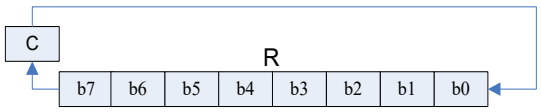
RETIE	Return from Interrupt and Enable Interrupt Globally
Syntax:	RETIE
Operand:	--
Operation:	Top of Stack→PC 1→GIE
Status affected:	--
Description:	The PC is loaded from top of Stack as return address and GIE is set to 1.
Cycle:	2
Example:	RETIE before executing instruction: GIE=0, Stack level=2. after executing instruction: GIE=1, PC=Stack[2], Stack pointer=1.

NOP	No Operation
Syntax:	NOP
Operand:	--
Operation:	No operation.
Status affected:	--
Description:	No operation.
Cycle:	1
Example:	NOP before executing instruction: PC=A0 after executing instruction: PC=A0+1

RETIA	Return with Data in ACC
Syntax:	RETIA i
Operand:	0 ≦ i < 255
Operation:	i→ACC, Top of Stack→PC
Status affected:	--
Description:	ACC is loaded with 8-bit immediate data i and PC is loaded from top of Stack as return address.
Cycle:	2
Example:	RETIA i before executing instruction: Stack pointer =2. i=0x55, ACC=0xAA. after executing instruction: PC=Stack[2], Stack pointer =1. ACC=0x55.

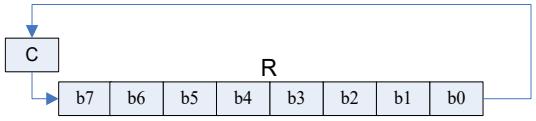
RET	Return from Subroutine
Syntax:	RET
Operand:	--
Operation:	Top of Stack→PC
Status affected:	--
Description:	PC is loaded from top of Stack as return address.
Cycle:	2
Example:	RET before executing instruction: Stack level=2. after executing instruction: PC=Stack[2], Stack level=1.

RLR	Rotate Left R Through Carry
Syntax:	RLR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	C→dest[0], R[7]→ C, R[6:0]→ dest[7:1]



Status affected:	C
Description:	The content of R is rotated one bit to the left through flag Carry. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	RLR R, d before executing instruction: R=0xA5, d=1, C=0. after executing instruction: R=0x4A, C=1.

RRR	Rotate Right R Through Carry
Syntax:	RRR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	C→dest[7], R[7:1]→ dest[6:0], R[0]→ C



Status affected:	C
Description:	The content of R is rotated one bit to the right through flag Carry. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	RRR R, d before executing instruction: R=0xA5, d=1, C=0. after executing instruction: R=0x52, C=1.

SBCAR	Subtract ACC and Carry from R
Syntax:	SBCAR R, d
Operand:	0 ≦ R ≦ 511 d = 0, 1.
Operation:	R + (~ACC) + C→dest
Status affected:	Z, DC, C
Description:	Subtract ACC and Carry from R with 2's complement representation. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	SBCAR R, d (a) before executing instruction: R=0x05, ACC=0x06, d=1, C=0, after executing instruction: R=0xFE, C=0. (-2) (b) before executing instruction: R=0x05, ACC=0x06, d=1, C=1, after executing instruction: R=0xFF, C=0. (-1) (c) before executing instruction: R=0x06, ACC=0x05, d=1, C=0, after executing instruction: R=0x00, C=1. (-0), Z=1. (d) before executing instruction: R=0x06, ACC=0x05, d=1, C=1, after executing instruction: R=0x1, C=1. (+1)

SBCIA	Subtract ACC and Carry from Immediate
Syntax:	SBCIA i
Operand:	0 ≦ i < 255
Operation:	i + (~ACC) + C→dest
Status affected:	Z, DC, C
Description:	Subtract ACC and Carry from 8-bit immediate data i with 2's complement representation. The result is placed in ACC.
Cycle:	1
Example:	SBCIA i (a) before executing instruction: i=0x05, ACC=0x06, C=0, after executing instruction: ACC=0xFE, C=0. (-2) (b) before executing instruction: i=0x05, ACC=0x06, C=1, after executing instruction: ACC=0xFF, C=0. (-1) (c) before executing instruction: i=0x06, ACC=0x05, C=0, after executing instruction: ACC=0x00, C=1. (-0), Z=1. (d) before executing instruction: i=0x06, ACC=0x05, C=1, after executing instruction: ACC=0x1, C=1. (+1)

SLEEP	Enter Halt Mode
Syntax:	SLEEP
Operand:	--
Operation:	00h→WDT, 00h→WDT prescaler 1→/TO 0→/PD
Status affected:	/TO, /PD
Description:	WDT and Prescaler0 are clear to 0. /TO is set to 1 and /PD is clear to 0. IC enter Halt mode.
Cycle:	1
Example:	SLEEP before executing instruction: /PD=1, /TO=0. after executing instruction: /PD=0, /TO=1.

SUBIA	Subtract ACC from Immediate
Syntax:	SUBIA i
Operand:	$0 \leqq i < 255$
Operation:	i - ACC→ACC
Status affected:	Z, DC, C
Description:	Subtract ACC from 8-bit immediate data i with 2's complement representation. The result is placed in ACC.
Cycle:	1
Example:	SUBIA i (a) before executing instruction: i=0x05, ACC=0x06. after executing instruction: ACC=0xFF, C=0. (-1) (b) before executing instruction: i=0x06, ACC=0x05, d=1, after executing instruction: ACC=0x01, C=1. (+1)

SUBAR	Subtract ACC from R
Syntax:	SUBAR R, d
Operand:	$0 \leqq R \leqq 511$ d = 0, 1.
Operation:	R - ACC→dest
Status affected:	Z, DC, C
Description:	Subtract ACC from R with 2's complement representation. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	SBCAR R, d (a) before executing instruction: R=0x05, ACC=0x06, d=1, after executing instruction: R=0xFF, C=0. (-1) (b) before executing instruction: R=0x06, ACC=0x05, d=1, after executing instruction: R=0x01, C=1. (+1)

SWAPR	Swap High/Low Nibble in R
Syntax:	SWAPR R, d
Operand:	$0 \leqq R \leqq 511$ d = 0, 1.
Operation:	R[3:0]→ dest[7:4]. R[7:4]→ dest[3:0]
Status affected:	--
Description:	The high nibble and low nibble of R is exchanged. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	SWAPR R, d before executing instruction: R=0xA5, d=1. after executing instruction: R=0x5A.

TABLEA	Read ROM data
Syntax:	TABLEA
Operand:	--
Operation:	ROM data{ TBHP, ACC } [7:0] ACC ROM data{TBHP, ACC} [15:8] TBHD.
Status affected:	--
Description:	The 8 least significant bits of ROM data pointed by {TBHP[3:0], ACC} is placed to ACC. The 8 most significant bits of ROM data pointed by {TBHP[3:0], ACC} is placed to TBHD[7:0].
Cycle:	2
Example:	TABLEA before executing instruction: TBHP = 0x02, ACC = 0x34. TBHD = 0x01. ROM data[0x234] = 0x35AA after executing instruction: TBHD=0x35, ACC=0xAA.

`

XORAR	Exclusive-OR ACC with R
Syntax:	XORAR R, d
Operand:	$0 \leqq R \leqq 511$ d = 0, 1.
Operation:	ACC⊕ R→dest
Status affected:	Z
Description:	Exclusive-OR ACC with R. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	XORAR R, d before executing instruction: R=0xA5, ACC=0xF0, d=1. after executing instruction: R=0x55.

XORIA	Exclusive-OR Immediate with ACC
Syntax:	XORIA i
Operand:	$0 \leqq i < 255$
Operation:	ACC⊕i→ACC
Status affected:	Z
Description:	Exclusive-OR ACC with 8-bit immediate data i. The result is stored in ACC.
Cycle:	1
Example:	XORIA i before executing instruction: i=0xA5, ACC=0xF0. after executing instruction: ACC=0x55.

24.Configuration Words

Item	Name	Options
1	High Oscillation Frequency	1. I_HRC 2. E_HXT 3. E_XT
2	Low Oscillation Frequency	1. I_LRC 2. E_LXT
3	High IRC Frequency	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz 7. 24MHz
4	High Crystal Oscillator	1. 6MHz ≦ E_HXT ＜ 8MHz 2. 8MHz ≦ E_HXT ＜ 10MHz 3. 10MHz ≦ E_HXT ＜ 12MHz 4. 12MHz ≦ E_HXT ＜ 16MHz 5. 16MHz ≦ E_HXT ＜ 20MHz 6. 20MHz ≦ E_HXT ＜ 24MHz
5	Instruction Clock	1. 2T 2. 4T
6	WDT	1. Enable (Software control) 2. Disable (Always disable)
7	WDT Event	1. Reset 2. Interrupt
8	Timer0 Source	1. EX_CKI0 2. Low Oscillator (I_LRC/E_LXT)
9	PA.5	1. PA.5 is I/O 2. PA.5 is reset
10	PA.7	1. PA.7 is I/O 2. PA.7 is instruction clock output
11	IR Output Pin	1.PA.3 2. PB.1
12	Startup Time	1. 600us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
13	WDT Time Base	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
14	Noise Filter (High_EFT)	1. Enable 2. Disable
15	LVR Setting	1. Register Control 2. Register Control + Halt mode Off 3. Always On 4. Operation mode On + Halt mode Off
16	LVR Voltage	1. 1.8V 2. 2.0V 3. 2.2V 4. 2.4V 5. 2.7V 6. 3.0V 7. 3.3V
17	VDD Voltage	1. 3.0V 2. 4.5V 3. 5.0V
18	VREFH Pin	1. PA.0 2. PB.1
19	Sink current type	1. Normal 2. Large 3. Ultra
20	Drive current type PA[7:4]	1. Normal 2. Large 3. Ultra
21	Drive current type	1. Normal 2. Large
22	Comparator/OPA Input pin select	1. Enable 2. Disable
23	Read Output Data	1. I/O Port 2. Register
24	E_LXT Backup Control	1. Auto Off 2. Register Off
25	EX_CKI0 to Inst. Clock	1. Sync 2. Async
26	Startup Clock	1. Fast (I_HRC/E_HXT/E_XT) 2. Slow (I_LRC/E_LXT)
27	Input Voltage Schmitt Trigger	1. Enable 2. Disable (0.5VDD)
28	Input High Voltage (V_{IH})	1. (0.7VDD) 2. (0.5VDD)

Item	Name	Options
29	Input Low Voltage (V_{IL})	1. (0.3VDD) 2. (0.2VDD)
30	INT0 Input Pin	1. PA.4 2.PB.0 3.PB.4
31	INT1 Input Pin	1. PA.3 2. PB.1
32	INT2 Input Pin	1. PA.5 2. Disable
33	UART Pin	1. PB.7 / PB.6 2. PB.0 / PB.1
34	I^2C Pin	1. PB.4 / PB.5 2. PC.2 / PC.3
35	EX_CKI1 Input Pin	1. PA.1 2. PA.2
36	PWM1 Output Pin	1. PA.3 2. PA.5 3. PB.1 4. PB.5
37	PWM2 Output Pin	1. PA.4 2 .PA.7 3. PB.3 4. PB.4 5. PB.7
38	PWM3 Output Pin	1. PA.2 2. PA.6 3. PB.0 4. PB.6
39	PWM4 Output Pin	1. PA.1 2. PA.4 3. PB.3 4. PC.1 5.PC.3
40	PWM5 Output Pin	1. PA.0 2. PB.2 3. PC.0 4. PC.2 5.PC.5
41	CCP1 pin (P1A / P1B / P1C / P1D)	1. PB.2 / PA.5 / PA.2 / PA.3 2. PC.4 / PA.1 / PA.2 / PA.3 3. PA.7 / PA.6 / PA.5 / PA.4
42	Comparator Input Pin	1. PA.0/PA.1/PA.2/PA.3 2. PA.2/PA.3/PA.4/PB.3
43	F_{VHRC} Frequency	1. 13.6MHz 2. 16MHz 3. 20.8MHz 4. 32MHz
44	EEPROM write MODE	1. One Byte 2. Continuous Write
45	ISP	1. Enable 2. Disable

25.Electrical Characteristics

25.1 Absolute Maximum Rating

Symbol	Parameter	Rated Value	Unit
V_{DD} - V_{SS}	Supply voltage	-0.5 ~ +6.0	V
V_{IN}	Input voltage	V_{SS}-0.3V ~ V_{DD}+0.3	V
T_{OP}	Operating Temperature	-40 ~ +85	˚C
T_{ST}	Storage Temperature	-40 ~ +125	˚C

25.2 DC Characteristics

(All refer F_{INST}=F_{HOSC}/4, F_{HOSC}=16MHz@I_HRC, WDT enabled, ambient temperature T_A=25˚C unless otherwise specified.)

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
V_{DD}	Operating voltage	--	2.7	--	5.5	V	F_{INST}=10MHz @ I_HRC 20MHz/2T
			2.4				F_{INST}=8MHz @ I_HRC 16MHz/2T
			2.0				F_{INST}=6MHz @ I_HRC 24MHz/4T
			2.0				F_{INST}=5MHz @ I_HRC 20MHz/4T
			1.8				F_{INST}=4MHz @ I_HRC 16MHz/4T
			1.8				F_{INST}=2MHz @ I_HRC 8MHz /4T
			1.8				F_{INST}=2MHz @ E_HXT 8MHz/4T
			1.8				F_{INST}=1MHz @ I_HRC 4MHz/4T
			1.8				F_{INST}=1MHz @ E_XT 4MHz/4T
V_{IH}	Input high voltage	5V	4.0	--	--	V	RSTb (0.8 V_{DD})
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins, EX_CKI0/1, INT0/1/2 (0.7 V_{DD})
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EX_CKI0/1 (0.5 V_{DD})
		3V	1.5	--	--		
V_{IL}	Input low voltage	5V	--	--	1.0	V	RSTb (0.2 V_{DD})
		3V	--	--	0.6		
		5V	--	--	1.5	V	All other I/O pins, EX_CKI0/1, INT0/1/2 (0.3 V_{DD})
		3V	--	--	0.9		
		5V	--	--	1.0	V	All other I/O pins, EX_CKI0/1 (0.2 V_{DD})
		3V	--	--	0.6		
I_{OH}	Output high current (Normal drive)	5V	--	20	--	mA	V_{OH}=4.0V
		3V	--	12	--		V_{OH}=2.0V
	Output high current (Larger drive)	5V	--	30	--	mA	V_{OH}=4.0V
		3V	--	20	--		V_{OH}=2.0V
	Output high current (Ultra drive)	5V	--	80	--	mA	V_{OH}=4.0V, @PA4-PA7
		3V	--	50	--		V_{OH}=2.0V, @PA4-PA7
I_{OL}	Output low current (Normal sink)	5V	--	26	--	mA	V_{OL}=1.0V
		3V	--	14	--		
	Output low current (Large sink)	5V	--	38	--	mA	V_{OL}=1.0V
		3V	--	21	--		
	Output low current (Ultra sink)	5V	--	65	--	mA	V_{OL}=1.0V
		3V	--	40	--		

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
I_{OP}	Operating current	**Normal Mode**					
		5V	--	2.7	--	mA	F_{INST} =10MHz @ I_HRC 20MHz/2T
		3V	--	1.7	--		
		5V	--	2.2	--	mA	F_{INST} =8MHz @ I_HRC 16MHz/2T
		3V	--	1.4	--		
		5V	--	2.1	--	mA	F_{INST} =6MHz @ I_HRC 24MHz/4T
		3V	--	1.3	--		
		5V	--	1.8	--	mA	F_{INST} =5MHz @ I_HRC 20MHz/4T
		3V	--	1.2	--		
		5V	--	2.9	--	mA	F_{INST} =5MHz @ E_HXT 20MHz/4T
		3V	--	1.4	--		
		5V	--	1.5	--	mA	F_{INST} =4MHz @ I_HRC 16MHz/4T
		3V	--	1.0	--		
		5V	--	2.4	--	mA	F_{INST} =4MHz @ E_HXT 16MHz/4T
		3V	--	1.0	--		
		5V	--	1.0	--	mA	F_{INST} =2MHz @ I_HRC 8MHz/4T
		3V	--	0.7	--		
		5V	--	1.8	--	mA	F_{INST} =2MHz @ E_HXT 8MHz/4T
		3V	--	0.7	--		
		5V	--	0.7	--	mA	F_{INST} =1MHz @ I_HRC 4MHz/4T
		3V	--	0.5	--		
		5V	--	0.6	--	mA	F_{INST} =1MHz @ E_XT 4MHz/4T
		3V	--	0.3	--		
		Slow Mode					
		5V	--	12.4	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/2, LVR ON
		3V	--	8.5	--		
		5V	--	14.0	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ E_LXT/2, LVR ON
		3V	--	8.8	--		
		5V	--	10.8	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4, LVR ON
		3V	--	7.6	--		
		5V	--	12.5	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ E_LXT/4, LVR ON
		3V	--	7.9	--		
I_{STB}	Standby current	5V	--	4.3	--	uA	Standby mode, F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4
		3V	--	2.0	--		
		5V	--	6.3	--	uA	Standby mode, F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4, LVR ON
		3V	--	3.6	--	uA	
I_{HALT}	Halt current	5V	--	0.5	5.0	uA	Halt mode, WDT disabled, LVR OFF.
		3V	--	0.4	3.0		
		5V	--	3.4		uA	Halt mode, WDT enabled, LVR OFF.
		3V	--	1.5			
		5V	--	2.9		uA	Halt mode, WDT disabled, LVR ON.
		3V	--	2.4			

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
R_{PH}	Pull-High resistor	5V	--	60	--	KΩ	Pull-High resistor
		3V	--	110	--		
R_{PL}	Pull-Low resistor	5V	--	60	--	KΩ	Pull-Low resistor
		3V	--	100	--		

25.3 Comparator / LVD Characteristics

(V_{DD}=5V, V_{SS}=0V, T_A=25°C unless otherwise specified.)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{IVR}	Comparator input voltage range	0	--	5	V	F_{HOSC}=1MHz
T_{ENO}	Comparator enable to output valid	--	20	--	uS	F_{HOSC}=1MHz
I_{CO}	Operating current of comparator	--	135	--	uA	F_{HOSC}=1MHz, P2V mode
I_{LVD}	Operating current of LVD	--	150	--	uA	F_{HOSC}=1MHz, LVD=4.15V
E_{LVD}	LVD voltage error	--	5	--	%	F_{HOSC}=1MHz, LVD=4.15V

25.4 OSC Characteristics

(Measurement conditions VDD Voltage, TA Temperature are equal to programming conditions.)

Parameter	Min.	Typ.	Max.	Unit	Condition
I_HRC deviation by socket			±1	%	Socket installed directly on writer.
I_HRC deviation by handler			±3	%	Handler condition with correct setup.
I_LRC deviation by handler			±5	%	

25.5 ADC Characteristics

(V_{DD}=5V, V_{SS}=0V, T_A=25°C unless otherwise specified.)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{REFH}	VREFH input voltage	2V	--	V_{DD}	V	Ext. reference voltage
V_{REF4}	Int. 4V reference voltage, V_{DD}=5V	3.94	4	4.06	V	
V_{REF3}	Int. 3V reference voltage, V_{DD}=5V	2.955	3	3.045	V	
V_{REF2}	Int. 2V reference voltage, V_{DD}=5V	1.97	2	2.03	V	
V_{REF}	Int. V_{DD} reference voltage, V_{DD}=5V	--	V_{DD}	--	V	
	Internal reference supply voltage	V_{REF}+0.5	--	--	V	Minimum supply voltage
	ADC analog input voltage	0	--	V_{REFH}	V	
	ADC enable time	256	--	--	us	Ready to start convert after set ADENB="1".

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
$I_{OP(ADC)}$	ADC current consumption	--	0.3	--	mA	
ADCLK	ADC Clock Frequency	--	--	2M	Hz	
ADCYCLE	ADC Conversion Cycle Time	16	--		1/ADCLK	SHCLK=2 ADC clock
ADC_{sample}	ADC Sampling Rate	--	--	125	K/sec	V_{DD}=5V
DNL	Differential Nonlinearity	±1	--	--	LSB	V_{DD}=5.0V, AVREFH=5V, FADSMP=62.5K
INL	Integral Nonlinearity	±2	--	--	LSB	
NMC	No Missing Code	10	11	12	Bits	

25.6 Flash/EEPROM Characteristics (-40 ~ 85° C)

*V_{LVR} is the abbreviation of LVR setting voltage.

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
$V_{DD-READ}$	Flash/EE Read Voltage	V_{LVR}	-	5.5	V	
$V_{DD-WRITE}$	Flash Write Voltage	2.4/V_{LVR}	-	5.5	V	
	EEPROM Write Voltage	1.8/V_{LVR}	-	5.5	V	
T_{WRITE}	Flash 16-bit Program Time (change code)	-	4.5	-	ms	
	Flash 16-bit Program Time (non-change code)	-	4.5	-	ms	
	EEPROM Page Program Time (change code)	-	4.5	-	ms	
	EEPROM Page Program Time (non-change code)		4,5		ms	
	EEPROM Word Program Time (change code)	-	4.5	-	ms	
	EEPROM Word Program Time (non-change code)	-	4.5	-	ms	
T_{RET}	Flash Data Hole Time	10	-	-	Year	
	EEPROM Data Hole Time	10	-	-	Year	
N_{END}	Flash Erase/Program Times	10k	-	-	Cycle	
	EEPROM Erase/Program Times	10k	-	-	Cycle	

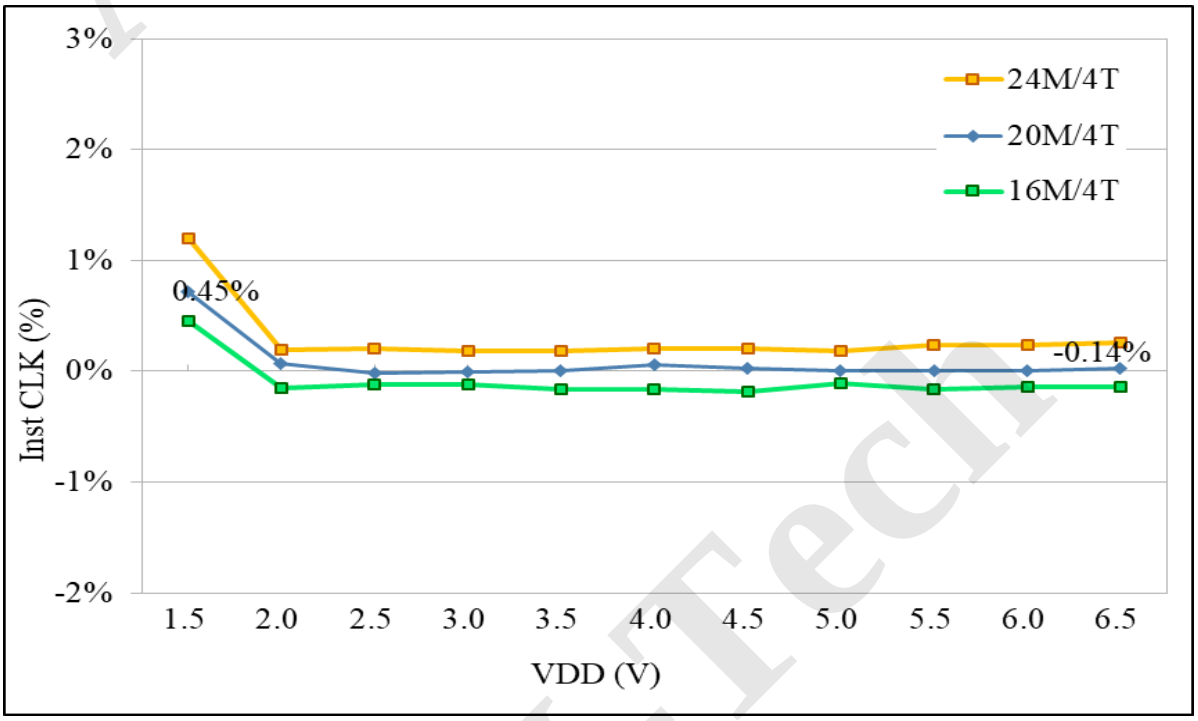
25.7 OPA Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{DD}	OPA Operating Voltage	2.5	-	5.5	V	
I_{DD}	OPA Operating Current	-	376	-	uA	VDD=5V
		-	303	-	uA	VDD=3V
V_{OS}	Input Offset Voltage	-	1	-	mV	0.5VDD
V_{CM}	Input Common Mode Voltage	0		VDD-1.5	V	
PSRR	Power Supply Rejection Ratio	60	70	-	dB	
AOL	Open Loop Gain	90	100		dB	

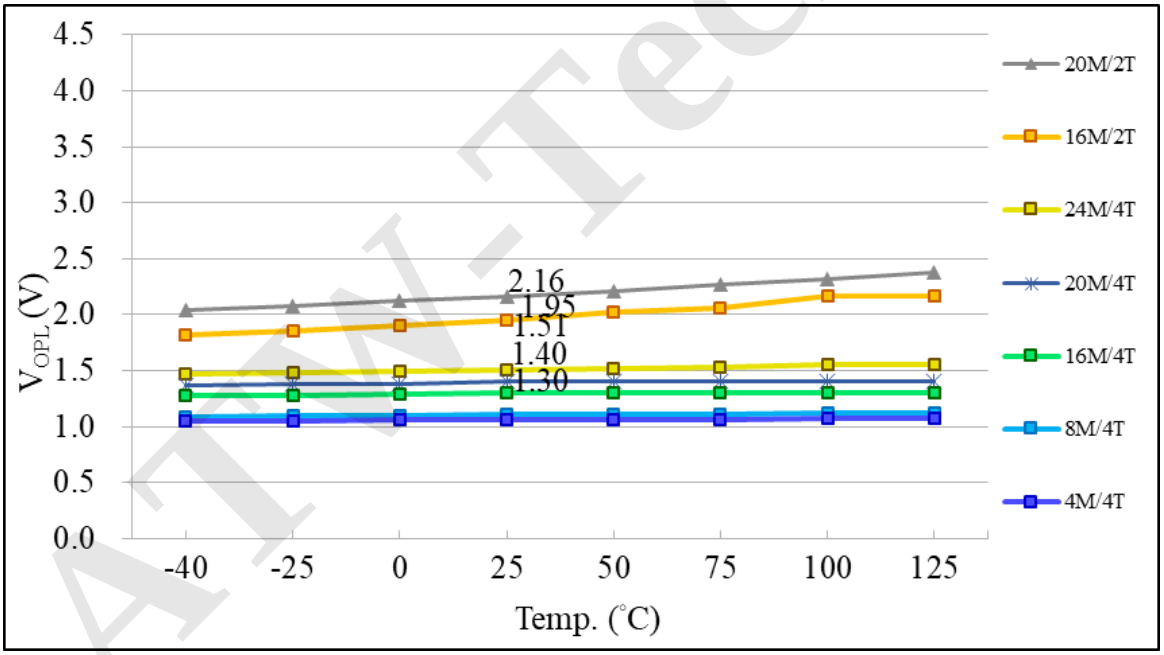
Note: Design assurance.

25.8 Characteristic Graph

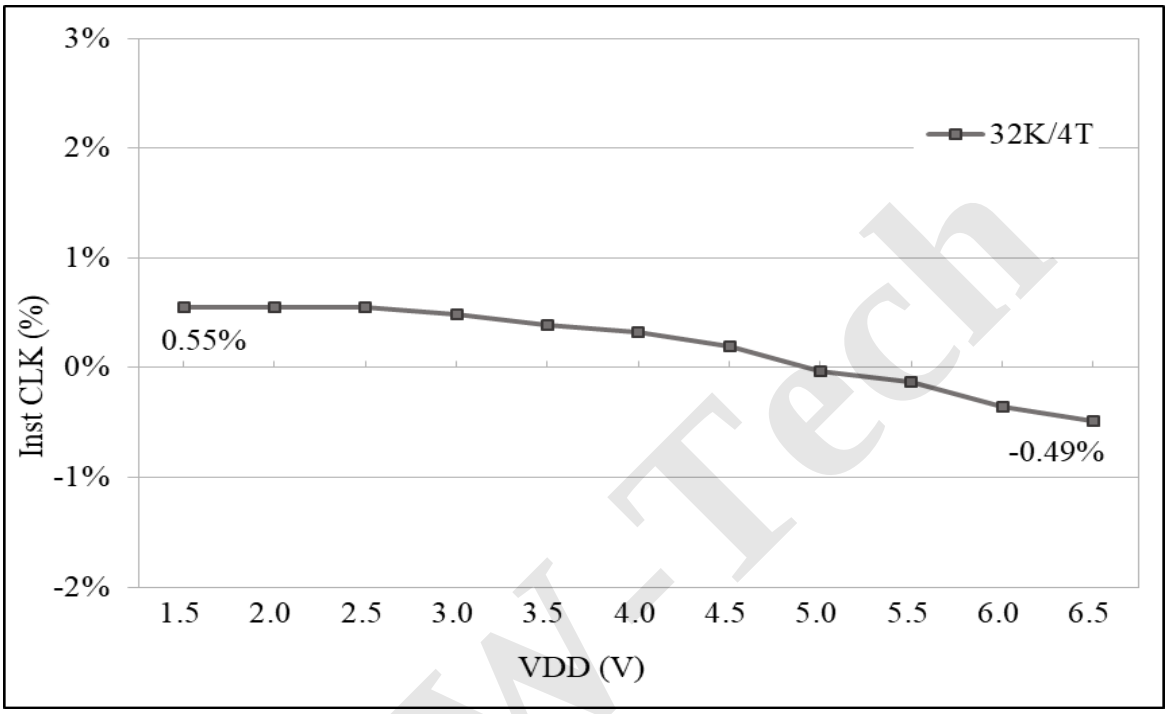
25.8.1 Frequency vs. V_{DD} of I_HRC



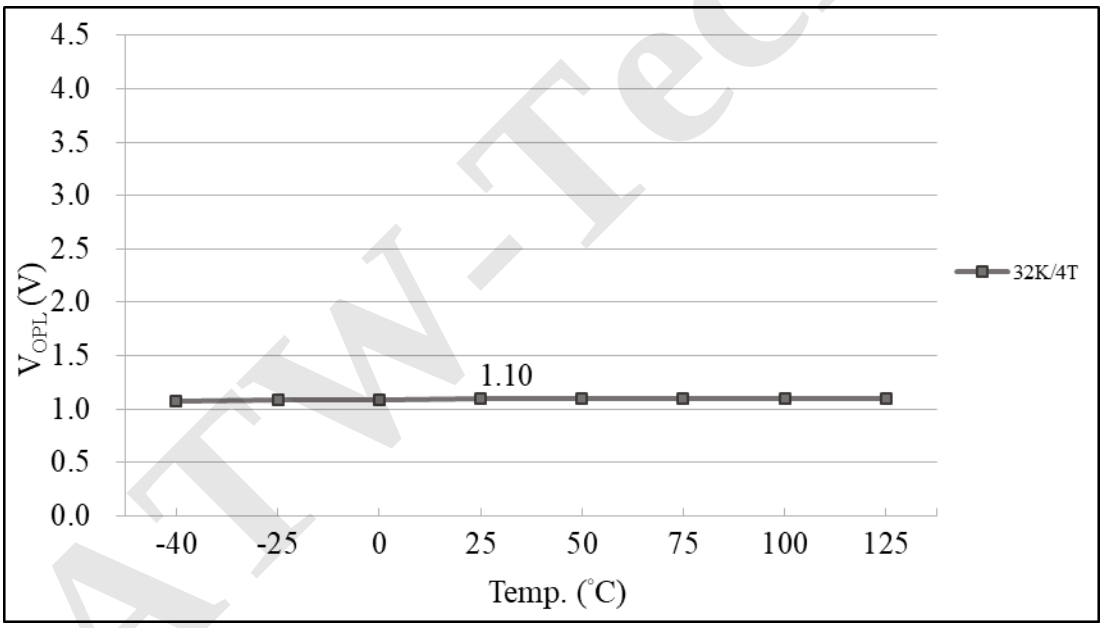
25.8.2 Frequency vs. Temperature of I_HRC



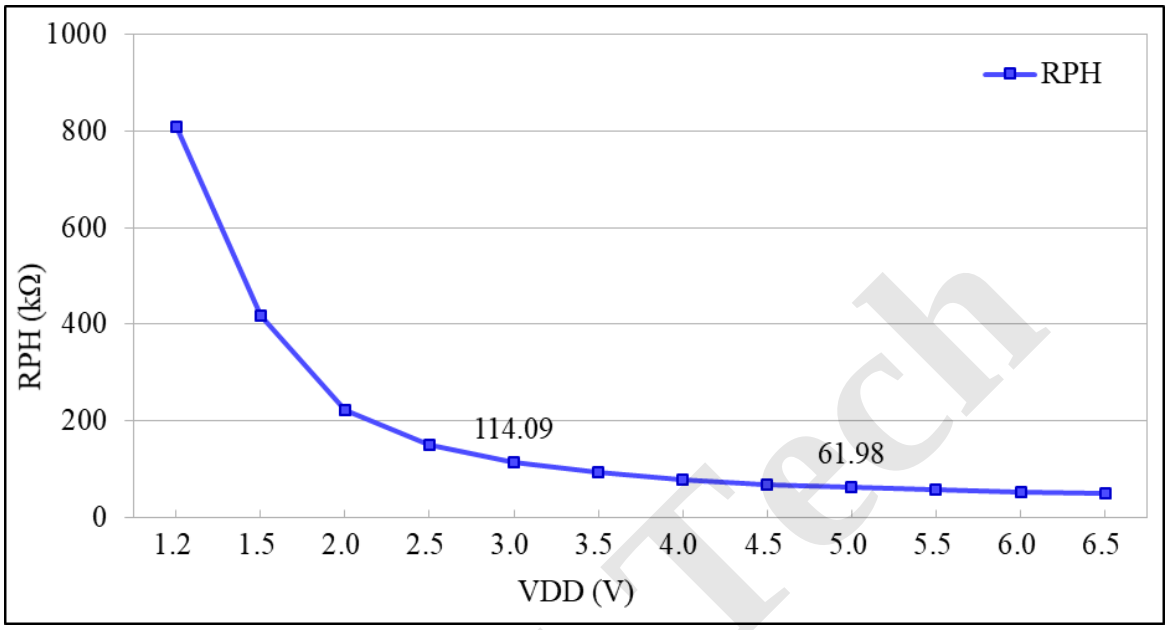
25.8.3 Frequency vs. V_{DD} of I_LRC



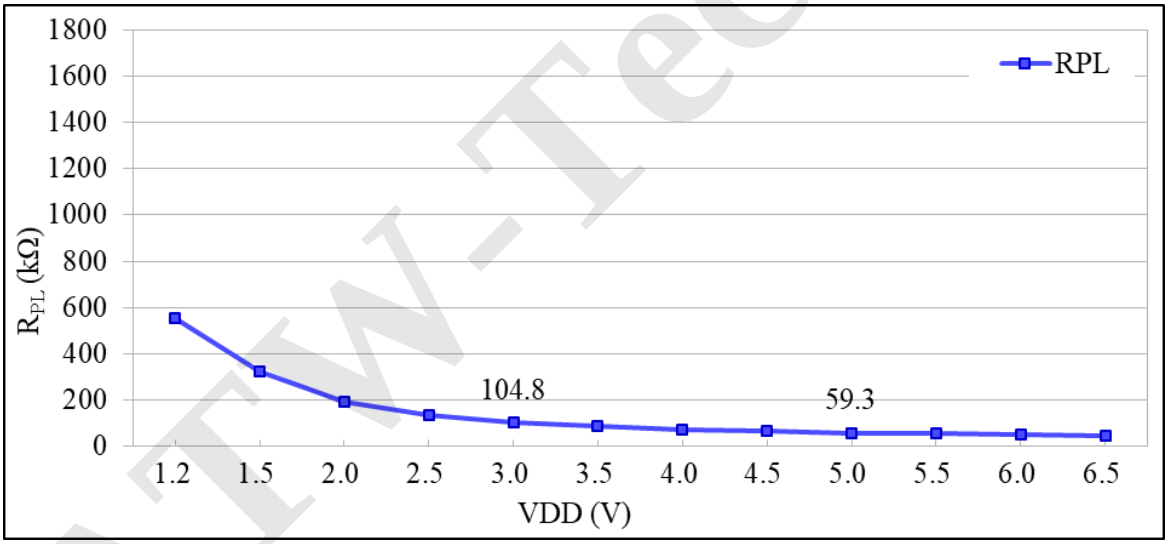
25.8.4 Frequency vs. Temperature of I_LRC



25.8.5 Pull High Resistor vs. V_{DD}



25.8.6 Pull Low Resistor vs. V_{DD}

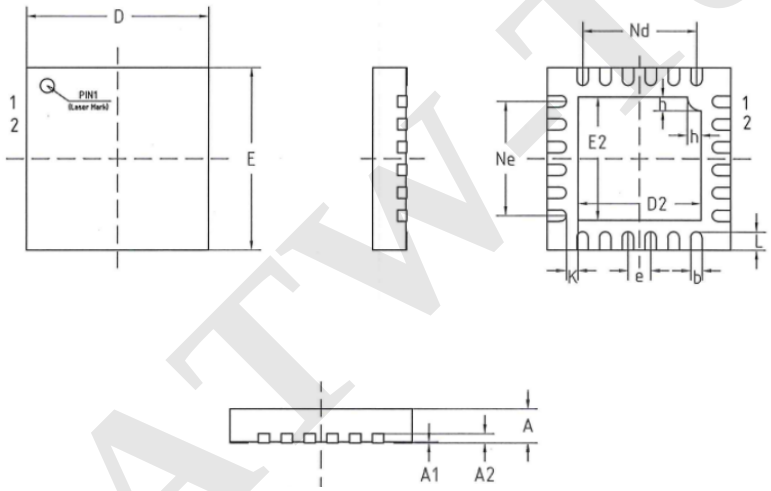


25.9 Recommended LVR Setting

Frequency	LVR default Setting	LVR Setting (Min. @25℃)
20M/2T	3.0V	2.7V
16M/2T	2.4V	2.4V
24M/4T	2.0V	2.0V
20M/4T	2.0V	2.0V
16M/4T	1.8V	1.8V
8M/4T	1.8V	1.8V
4M/4T	1.8V	1.8V

26 Package Dimension

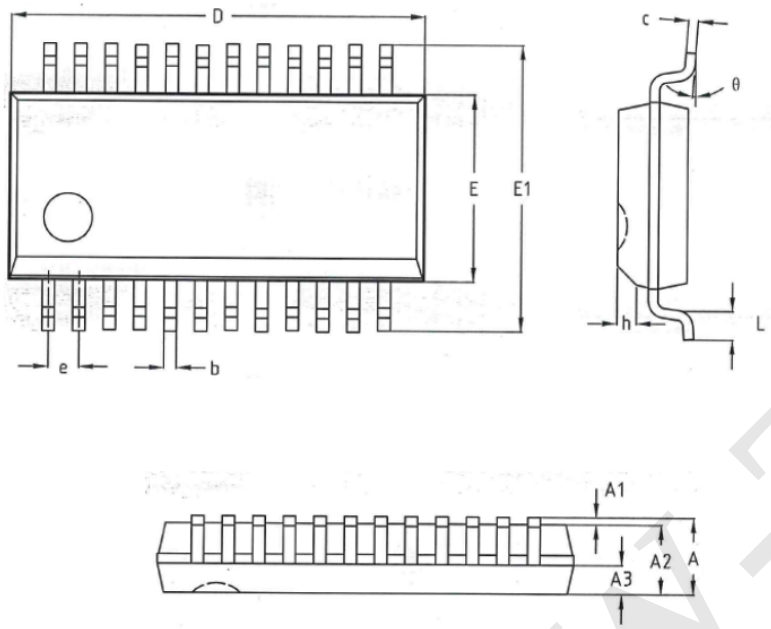
26.1 24-Pin Plastic QFN



	MILLIMETERS		
SYMBOL	MIN	TYP	MAX
A	0.70	0.75	0.80
A1	-	0.02	0.05
A2	0.203 REF		
b	0.20	0.25	0.30
D	3.90	4.00	4.10
D2	2.60	2.70	2.80
E	3.90	4.00	4.10
E2	2.60	2.70	2.80
e	0.50 BSC		
K	0.20	0.25	0.30
L	0.35	0.40	0.45
h	0.25	0.30	0.35
Ne	2.50 BSC		
Nd	2.50 BSC		

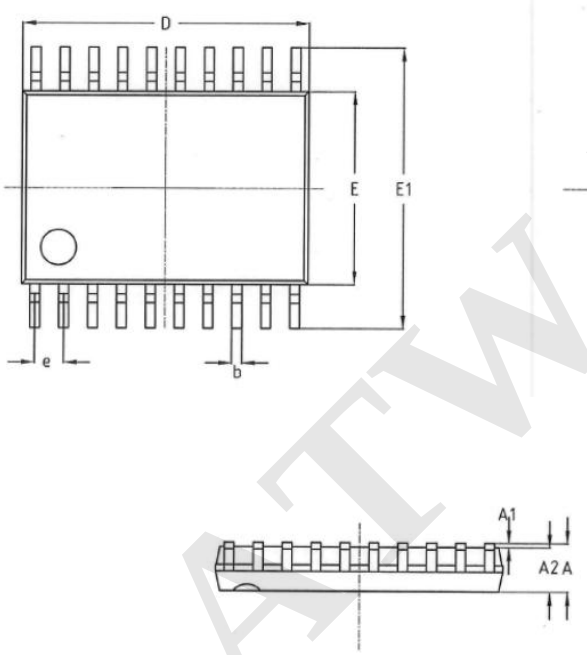
Note: For 20-Pin SOP, 35 units per tube.

26.2 24-Pin Plastic SSOP



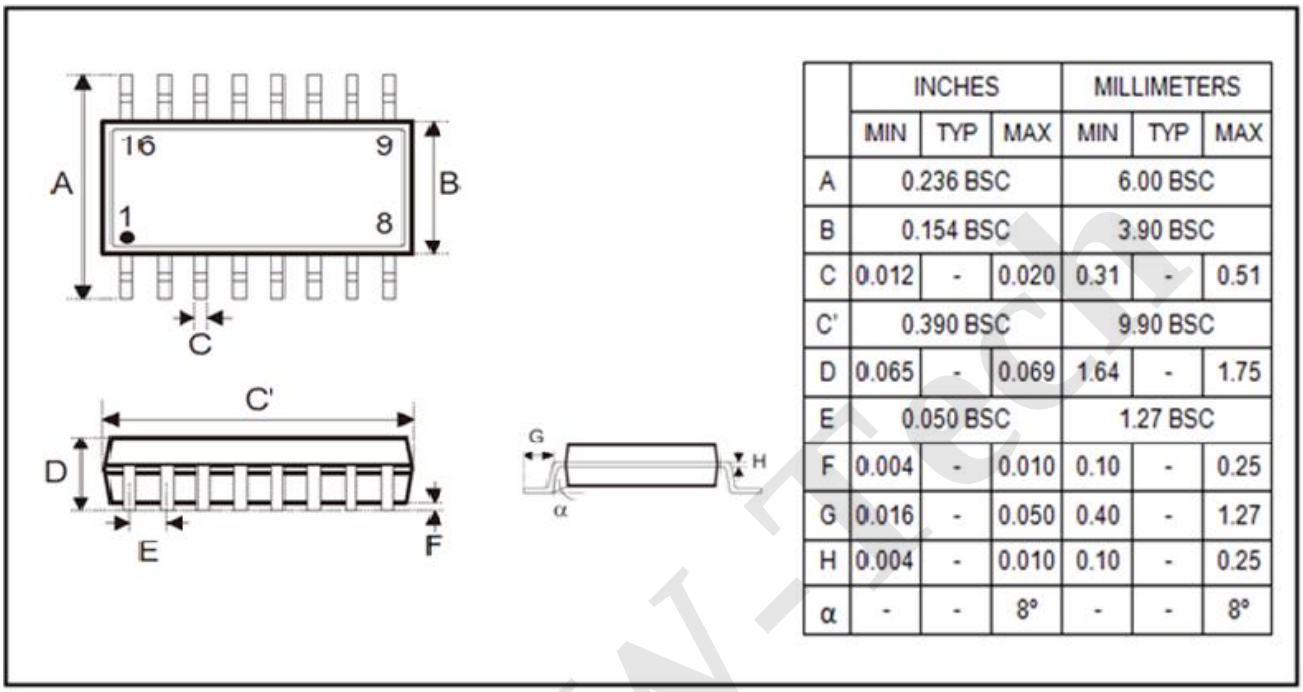
	MILLIMETERS		
SYMBOL	MIN	TYP	MAX
A	-	-	1.75
A1	0.10	-	0.25
A2	1.35	1.45	1.55
A3	0.60	0.65	0.70
b	0.23	-	0.31
C	0.19	-	0.25
D	8.50	8.60	8.70
E	3.80	3.90	4.00
E1	5.80	6.00	6.20
e	0.635 BSC		
L	0.30	-	0.50
h	0.40	-	0.80
θ	0°	-	8°

26.3 20-Pin Plastic TSSOP



SYMBOL	MILLIMETERS		
	MIN	TYP	MAX
A	-	-	1.20
A1	0.05	-	0.15
A2	0.90	1.00	1.05
b	0.19	-	0.30
C	0.09	-	0.20
D	6.40	6.50	6.60
E	4.30	4.40	4.50
E1	6.25	6.40	6.55
e	0.65 BSC		
L1	1.00REF		
L	0.45	0.60	0.75
θ	0°	-	8°

26.4 16-Pin Plastic SOP (150 mil)



	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.236 BSC			6.00 BSC		
B	0.154 BSC			3.90 BSC		
C	0.012	-	0.020	0.31	-	0.51
C'	0.390 BSC			9.90 BSC		
D	0.065	-	0.069	1.64	-	1.75
E	0.050 BSC			1.27 BSC		
F	0.004	-	0.010	0.10	-	0.25
G	0.016	-	0.050	0.40	-	1.27
H	0.004	-	0.010	0.10	-	0.25
α	-	-	8°	-	-	8°

27. Ordering Information

P/N	Package Type	Pin Count	Package	Shipping
AT8F2481S16	SOP	16	150 mil	Tube: 50 pcs per Tube
AT8F2481T20	TSSOP	20	-	Tape & Reel: 4K pcs per Reel Tube:70 pcs per Tube
AT8F2481U24	SSOP	24	-	Tape & Reel: 2.5K pcs per Reel Tube: 50 pcs per Tube
AT8F2481NA24	QFN	24	QFN24L (4x4x0.75-P0.5)	Tape & Reel: 4K pcs per Reel